INSTALLATION AND USER'S GUIDE

F5 LABS

Service and Support Information

Product Version

This manual applies to version 2.0.1 of the BIG/ip Controller platform, including the BIG/ip HA, BIG/ip HA+, and BIG/ip LB Controllers. To obtain technical support for these products, or to request product sales or customer service information, refer to the contact information provided below.

Telephone

Corporate: (206) 505-0800

Corporate toll-free: (888) 88BIG-IP

Technical Help Line: (206) 505-0888

Fax: (206) 505-0801

Mailing Address

200 1st Avenue West
Suite 500
Seattle, Washington 98119

Electronic Mail

Technical Help: support@f5.com

Sales Information: sales@f5.com

Product feedback: feedback@f5.com

World Wide Web

www.f5.com

Legal Notices

Copyright

Trademarks

Export Regulation Notice

The BIG/ip Controller is shipped with cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this BIG/ip Controller from the United States.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a

commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with this instruction manual, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

Acknowledgments

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by Charles Hannum.

This product includes software developed by Charles Hannum, by the University of Vermont and Stage Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD," "FreeBSD," "Mach" (by CMU).

In the following statement, "This software" refers to the parallel port driver: This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.

F5 Labs Limited Warranty

This warranty will apply to any sale of goods or services or license of software (collectively, "Products") from F5 Labs, Inc. ("F5"). Any additional or different terms including terms in any purchase order or order confirmation will have no effect unless expressly agreed to in writing by F5. Any software provided to a Customer is subject to the terms of the End User License Agreement delivered with the Product.

Limited Warranty

Software. F5 warrants that for a period of 90 days from the date of shipment: (a) the media on which the software is furnished will be free of defects in materials and workmanship under normal use; and (b) the software substantially conforms to its published specifications. Except for the foregoing, the software is provided AS IS.

In no event does F5 warrant that the Software is error free, that the Product will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Product will satisfy Purchaser's own specific requirements.

Hardware. F5 warrants that the hardware component of any Product will, for a period of one year from the date of shipment from F5, be free from defects in material and workmanship under normal use.

Remedy. Purchaser's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any Product or component that fails during the warranty period at no cost to Purchaser. Products returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of the package, and sent prepaid, insured, and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Purchaser, at F5's expense, no later than 7 days after receipt by F5. Title to any returned Products or components will transfer to F5

upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the software to correct any substantial non-conformance with the specifications.

Restrictions. The foregoing limited warranties extend only to the original Purchaser, and do not apply if a Product (a) has been altered, except by F5, (b) has not been installed, operated, repaired, or maintained in accordance with F5's instructions, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (d) has been operated outside of the environmental specifications for the Product. F5's limited software warranty does not apply to software corrections or upgrades.

Support, Upgrades. F5 provides software telephone support services at no charge for 90 days following the installation of any Product: Monday through Friday, from 6 a.m. to 6 p.m. Pacific time, excluding F5's holidays. Such support will consist of responding to trouble calls as reasonably required to make the Product perform as described in the Specifications. For advisory help requests, which are calls of a more consultative nature than a standard trouble call, F5 will provide up to two hours of telephone service at no charge. Additional service for advisory help requests may be purchased at F5 Labs' then-current standard service fee. During this initial 90 day period, Customer is entitled, at no charge, to updated versions of covered software such as bug fixes, and incremental enhancements as designated by minor revision increases (for example, BIG/ip V1.5 to BIG/ipV1.6). In addition, Customer will receive special pricing on upgraded versions of covered Products such as new clients, new modules, and major enhancements designated by major revision increases (for example, BIG/ip V1.x to BIG/ip V2.0). Customer may purchase a Maintenance Agreement for enhanced maintenance and support services.

DISCLAIMER; LIMITATION OF REMEDY: EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 DOES NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO PRODUCTS, SPECIFICATIONS, SUPPORT, SERVICE, OR ANYTHING ELSE. F5 HAS NOT AUTHORIZED ANYONE TO MAKE ANY REPRESENTATION OR WARRANTY OTHER THAN AS PROVIDED ABOVE. F5 DISCLAIMS ANY AND ALL

WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED, OR OTHERWISE, ARISING WITH RESPECT TO THE PRODUCTS OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. F5 WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE, OR IMPUTED NEGLIGENCE, STRICT LIABILITY, OR PRODUCT LIABILITY), OR OTHERWISE FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH ANY OF THE PRODUCTS OR OTHER GOODS OR SERVICES FURNISHED TO CUSTOMER BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

End-user Software License

IMPORTANT! READ BEFORE INSTALLING OR OPERATING THIS PRODUCT.

CAREFULLY READ THE TERMS AND CONDITIONS OF THIS LICENSE BEFORE INSTALLING OR OPERATING THIS PRODUCT: BY INSTALLING, OPERATING, OR KEEPING THIS PRODUCT FOR MORE THAN THIRTY DAYS AFTER DELIVERY, YOU INDICATE YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THE TERMS AND CONDITIONS OF THIS AGREEMENT, PROMPTLY CONTACT F5 LABS, INC. ("F5") TO ARRANGE FOR RETURN OF THE PRODUCT FOR A REFUND.

1. Scope. This License applies to the software for the BIG/ip Controller, whether such software is provided separately or as an integral part of a hardware product. As used herein, the term "Software" will refer to all such software, and the corrections, updates, new releases and new versions of such software. A product that consists of Software only will be referred to as a "Software Product" and a combination Software/Hardware product will be referred to as a "Combination Product." All Software is licensed, not sold, by F5. This License is a legal agreement between F5 and the single entity ("Licensee") that has acquired Software from F5 under applicable terms and conditions.

2. License Grant. Subject to the terms of this License, F5 grants to Licensee a non-exclusive, non-transferable license to use the Software in object code form solely on a single central processing unit owned or leased by Licensee. Other than as specifically described herein, no right or license is granted to Licensee to any of F5's trademarks, copyrights, or other intellectual property rights. Licensee may make one back-up copy of any Software Product, provided the back-up copy contains the same copyright and proprietary information notices as the original Software Product. Licensee is not authorized to copy the Software contained in

a Combination Product. The Software incorporates certain third party software which is used subject to licenses from the respective owners.

3. Restrictions. The Software, documentation, and the associated copyrights are owned by F5 or its licensors, and are protected by law and international treaties. Except as provided above, Licensee may not copy or reproduce the Software, and may not copy or translate the written materials without F5's prior, written consent. Licensee may not copy, modify, reverse compile, or reverse engineer the Software, or sell, sub-license, rent, or transfer the Software or any associated documentation to any third party.

4. Export Control. F5's standard Software incorporates cryptographic software. Licensee agrees to comply with the Export Administration Act, the Export Control Act, all regulations promulgated under such Acts, and all other laws and governmental regulations relating to the export of technical data, and equipment, and products produced therefrom, which are applicable to Licensee.

5. Limited Warranty.

   a) Warranty. F5 warrants that for a period of 90 days from the date of shipment: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications. Except for the foregoing, the Software is provided AS IS. In no event does F5 warrant that the Software is error-free, that it will operate with any software or hardware other than that provided by F5 or specified in the documentation, or that the Software will satisfy Licensee's own specific requirements.

   b) Remedy. Licensee's exclusive remedy and the entire liability of F5 under this limited warranty and any other guarantee made by F5 is, at F5's option, to repair or replace any F5 product that fails during the warranty period at no cost to Licensee. Any products returned to F5 must be pre-authorized by F5 with a Return Material Authorization (RMA) number marked on the outside of

the package, and sent prepaid, insured, and packaged appropriately for safe shipment. The repaired or replaced item will be shipped to Licensee, at F5's expense, no later than 7 days after receipt by F5. Title to any returned product or components will transfer to F5 upon receipt. F5 will replace defective media or documentation or, at its option, undertake reasonable efforts to modify the Software to correct any substantial non-conformance with the specifications.

c) Restrictions. The foregoing limited warranties extend only to the original Licensee, and do not apply if a Software Product or Combination Product (i) has been altered, except by F5, (ii) has not been installed, operated, repaired, or maintained in accordance with F5's instructions, (iii) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident or (iv) has been operated outside of the environmental specifications for the product. F5's limited software warranty does not apply to software corrections or upgrades.

6. DISCLAIMER; LIMITATION OF REMEDY. EXCEPT FOR THE WARRANTIES SPECIFICALLY DESCRIBED HEREIN, F5 DOES NOT MAKE ANY GUARANTEE OR WARRANTY, EXPRESS OR IMPLIED, WITH RESPECT TO THE SOFTWARE, SPECIFICATIONS, SUPPORT, SERVICE OR ANYTHING ELSE. F5 HAS NOT AUTHORIZED ANYONE TO MAKE ANY REPRESENTATION OR WARRANTY OTHER THAN AS PROVIDED ABOVE. F5 DISCLAIMS ANY AND ALL WARRANTIES AND GUARANTEES, EXPRESS, IMPLIED OR OTHERWISE, ARISING WITH RESPECT TO THE SOFTWARE OR SERVICES DELIVERED HEREUNDER, INCLUDING BUT NOT LIMITED TO THE WARRANTY OF MERCHANTABILITY, THE WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTY OF NON-INFRINGEMENT OF THE INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY. F5 WILL HAVE NO OBLIGATION OR LIABILITY, WHETHER ARISING IN

CONTRACT (INCLUDING WARRANTY), TORT (INCLUDING ACTIVE, PASSIVE, OR IMPUTED NEGLIGENCE, STRICT LIABILITY OR PRODUCT LIABILITY), OR OTHERWISE, FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR INDIRECT DAMAGES INCLUDING BUT NOT LIMITED TO LOSS OF USE, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF REVENUE, LOSS OF BUSINESS, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR OTHER GOODS OR SERVICES FURNISHED TO LICENSEE BY F5, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination. This License is effective until terminated, and will automatically terminate if Licensee fails to comply with any of its provisions. Upon termination of this License, the Licensee will destroy the Software and documentation and all copies or portions thereof.

8. Miscellaneous. This Agreement will be governed by the laws of the State of Washington, USA without regard to its choice of law rules. The provisions of the U.N. Convention for the International Sale of Goods will not apply. Any provisions found to be unenforceable will not affect the enforceability of the other provisions contained herein, but will instead be replaced with a provision as similar in meaning to the original as possible. This Agreement constitutes the entire agreement between the parties with regard to its subject matter. No modification will be binding unless in writing and signed by the parties.

Table of Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

1

Introduction to the BIG/ip Controller

- **The BIG/ip Server Array Controller**
- **Balancing and managing connections**
- **Working with configuration and monitoring tools**
- **What's new in version 2.0**

The BIG/ip Server Array Controller

The BIG/ip® Server Array Controller is a controller that manages and balances network traffic. A BIG/ip Controller can intelligently distribute site connections across arrays of servers, transparent firewalls, transparent cache servers, routers, as well as other router-like devices. The BIG/ip platform is designed to manage connections for multiple Internet or intranet sites, and it supports a wide variety of Internet protocols and services.

The BIG/ip platform also assures a consistently high level of server availability by continually monitoring several aspects of the network servers that deliver content for the site. A BIG/ip Controller can verify whether a server responds to a ping, whether a server allows the BIG/ip Controller to connect to a specific service, and whether specific site content is currently available on a particular server. A BIG/ip Controller never attempts to send connections to a server that is down or too busy to handle the connection.

The BIG/ip platform is highly configurable, and network administrators can use the BIG/config web application for easy system configuration and monitoring, as well as traditional command line utilities. Administrators can choose from seven different load balancing modes, and they can also take advantage of popular network administration tools such as Sendmail, and the Simple Network Management Protocol (SNMP).

Controlling network traffic for multiple sites

The BIG/ip platform actually controls and balances network traffic at the port level. A BIG/ip Controller can distribute site connections among multiple ports on an individual server, and it can balance traffic across multiple servers. Some sites may even use multiple ports on a single server to handle the same Internet service, thus expanding the throughput of that individual server for the given service.

A basic configuration

Figure 1.1 shows an example of a simple BIG/ip Controller configuration that manages two web sites. In this example, each site supports a specific Internet service, and each site stores content on two of the three servers in the array. Both sites store content on Server 2.

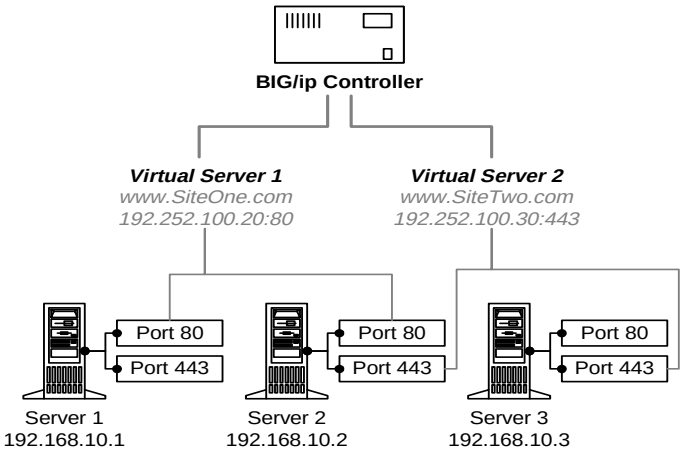

Figure 1.1 *A basic configuration*

A ***virtual server*** is a specific combination of a virtual IP address and virtual port number. Figure 1.1 shows two virtual servers. Virtual Server 1 is configured on the BIG/ip Controller, and it handles HTTP services for *www.SiteOne.com*. Virtual Server 1 is identified by the virtual IP address and virtual port number 198.252.100.20:80. Virtual Server 1 is mapped to two different physical IP address:port numbers, referred to as ***nodes***.

Virtual Server 2 is also configured on the BIG/ip Controller, and it handles SSL services for *www.SiteTwo.com*. Virtual Server 2 is also identified by a virtual IP address and virtual port number, 198.252.100.30:443, which is mapped to two different physical nodes.

The BIG/ip Controller distributes *www.SiteOne.com* connections to port 80 on two different physical servers: Server 1 (192.168.10.1) and Server 2 (192.168.10.2). The BIG/ip Controller distributes *www.SiteTwo.com* connections to port 443 on Server 2 (192.168.10.2) and Server 3 (192.168.10.3). Note that Server 2 (192.168.10.2) is used to support both web sites, providing HTTP service on port 80 for *www.SiteOne.com*, and also providing SSL service on port 443 for *www.SiteTwo.com*.

The BIG/ip Controller distributes connections among the three servers according to a user-specified load balancing algorithm.

Internet protocol and service support

The BIG/ip platform supports both TCP and UDP protocols, as well as the following popular Internet services:

- HTTP
- FTP (Active and Passive)
- SMTP
- NNTP
- POP
- DNS
- Real Audio/TCP
- IMAP
- Telnet

Network administrators should note that you can configure persistence settings for both TCP and UDP connections. You can also configure SSL persistence settings that can work in conjunction with TCP persistence settings.

Configuration scalability

The BIG/ip platform is designed to manage up to 10,000 virtual servers, though most common configurations are significantly smaller. The number of content servers that a BIG/ip Controller can load balance is limited only by the capacity of the network media, such as Ethernet, that sits between the BIG/ip Controller and

the servers. The maximum number of concurrent connections that a BIG/ip Controller can manage is determined by the amount of RAM in your particular BIG/ip hardware configuration. The BIG/ip platform offers a variety of hardware configurations, including BIG/ip HA and BIG/ip HA+. For information about specific configurations, refer to the technical specifications sheet supplied with your BIG/ip Controller.

Maintaining site reliability

When you incorporate a BIG/ip Controller system into your network, you gain consistent reliability in three important ways:

- **Distributed site content**
  Your site content is accessible on more than one server. The BIG/ip Controller allows you to take individual servers down for maintenance, and to return them to service without disrupting the flow of traffic.
- **Intelligent connection distribution**
  The BIG/ip Controller intelligently manages connections among multiple content servers, working to prevent server overload, and never attempting to send connections to servers that are not available. For example, the BIG/ip platform's ***Extended Content Verification*** feature not only verifies that a server is running; it also verifies that all the different processes involved in creating a dynamic web page are thoroughly checked before service requests are routed to the server.
- **Hardware redundancy**
  A BIG/ip redundant system provides two BIG/ip Controller units, one of which runs as an ***active*** system and manages all connections, while the other unit runs as a ***standby*** system. In the event that the active BIG/ip Controller goes down, the standby BIG/ip Controller immediately becomes the active unit and manages all connections without disrupting network service.

Making use of site verification options

The BIG/ip platform provides four different methods for verifying that site content servers are available.

- **Node ping**
  Node ping requires that the BIG/ip Controller send a standard echo ping to each server's IP address. If the server responds to the ping within a set time frame, the BIG/ip Controller determines the server to be available.
- **Service check**
  Service check requires that the BIG/ip Controller attempt to connect to a specific port, and verify that the service hosted by the port is available. If the BIG/ip Controller successfully establishes a conversation with the service, the server is considered available.
- **ECV service check**
  ECV service check uses the BIG/ip Controller Extended Content Verification feature to perform a sophisticated type of service check. Extended Content Verification requires that the BIG/ip Controller connect to a port, request specific data, such as text that is included in an HTML page, and then verify whether the server returned the requested data. If the server returns the requested data, the BIG/ip Controller considers the server to be available.
- **EAV service check**
  EAV service check uses the BIG/ip Controller Extended Application Verification feature to perform a sophisticated type of service check. An EAV service check is similar to an ECV service check, except that it allows a custom program, typically developed by the customer, to perform the content verification on behalf of the BIG/ip Controller software. If the program returns a positive result after performing the service check, the BIG/ip Controller considers the server to be available.

Working with BIG/ip redundant hardware systems

In a BIG/ip redundant system, two BIG/ip Controllers are connected by a fail-over cable. One of the two units serves as the active BIG/ip Controller, and it processes all connections. The other unit is a standby unit that is always prepared to become the active unit should a fail-over occur. Both units in the redundant system share an IP address, which ensures that in the event of a fail-over, network traffic is still routed to the appropriate machine.

Figure 1.2 shows a BIG/ip redundant hardware system that uses the shared IP address, 198.252.168.100.

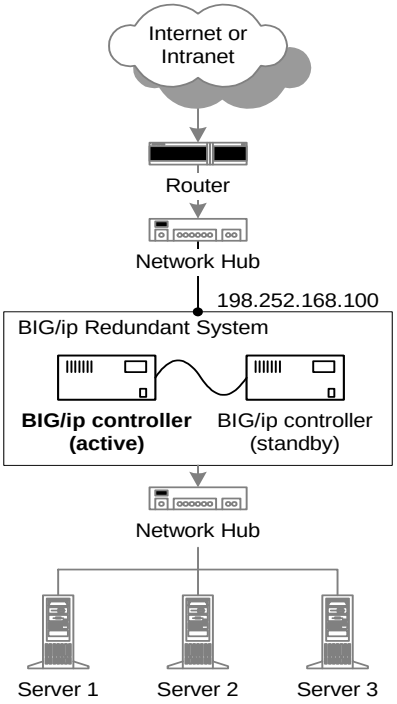

Figure 1.2 *A BIG/ip redundant system*

System fail-over

Fail-over is actually controlled by a watchdog timer card, which monitors BIG/ip hardware, and by BIG/ip software, which monitors various aspects of the BIG/ip Controller system. If the watchdog timer or the BIG/ip software on the active unit doesn't receive the expected responses from either the system hardware or the system software within a specified amount of time, a fail-over occurs. The active BIG/ip Controller automatically passes control to the standby unit. The standby BIG/ip Controller immediately becomes the

active unit and begins handling connections. The other BIG/ip Controller initiates a reboot sequence and becomes the standby unit, prepared to take over should another fail-over occur.

Balancing and managing connections

A key element of the BIG/ip Controller is its ability to balance and control the flow of traffic to individual ports on specific servers once the servers are verified as being available. The BIG/ip platform offers the following important features that help you control the balance and flow of network traffic among the servers in an array:

- Static and dynamic load balancing modes
- IP packet filtering and rate classes that control traffic speed
- Connection limits
- Configurable persistence modes for TCP, UDP, and SSL

Load balancing

The BIG/ip Controller offers seven different load balancing modes, including three static modes and four dynamic modes. A load balancing mode defines, in part, the logic that a BIG/ip Controller uses to determine which server should receive a particular connection on a specific port.

Static load balancing

Static load balancing is based on pre-defined user settings, and does not take current performance into account. The BIG/ip platform supports three static load balancing modes:

- **Round Robin**
  Round Robin mode is a basic load balancing mode that distributes connections evenly across all ports, passing each new connection to the next port in line.

- **Ratio**
  The Ratio mode distributes new connections across ports in proportion to a user-defined ratio. For example, if your array contained one new, high-speed server and two older servers, you could set the ratio so that the high-speed server receives twice as many connections as either of the two older servers.
- **Priority**
  The Priority mode distributes connections in round robin fashion to a specific groups of servers. It begins distributing new connections to the highest priority group. If all servers in that group should go down, it begins distributing connections to servers in the next higher priority group.

Dynamic load balancing

Dynamic load balancing modes use current performance information from each node to determine which node should receive each new connection. The different dynamic load balancing modes incorporate different performance factors:

- **Least Connections**
  In Least Connections mode, the BIG/ip Controller sends each new connection to the node that currently hosts the fewest current connections.
- **Fastest**
  In Fastest mode, the BIG/ip Controller sends each new connection to the node that has the fastest measured response time.
- **Observed**
  In Observed mode, the BIG/ip Controller sends each new connection to the node that has the highest performance rating, based on a combination of fewest connections and fastest response time.
- **Predictive**
  Predictive mode factors in both performance ratings as well as performance improvement over time.

IP packet filtering and rate classes

The BIG/ip platform supports easy configuration of BSD/OS IP packet filtering. IP packet filtering allows you to control both in-bound and out-bound network traffic. For example, you can specify a single IP address, or a range of IP addresses, from which your site either accepts or denies network traffic. You can also specify one or more IP addresses to which you specifically want to allow or prevent out-bound connections.

The BIG/ip platform also supports rate classes, which are an extension to IP filters. A rate class defines a maximum packet rate (bits per second) for connections that originate from a specific IP address or from a range of IP addresses. You can use rate classes to help control the amount and flow of specific network traffic. For example, you can offer faster connection speeds for high priority connections, such as paying customers on an e-commerce site.

Connection limits

The BIG/ip Controller allows you to set limits on connections in three different ways:

- Maximum number of concurrent connections accepted on a single port
- Minimum persistence time for connections that require persistence, such as UDP or SSL
- Maximum time for connections to remain idle before being dropped

Configurable persistence modes

The BIG/ip Controller provides support for TCP, UDP, and SSL persistence. The BIG/ip Controller allows you to set TCP and UDP persistence on the ports that it manages. You can set persistence for SSL connections on individual virtual servers (a specific combination of an IP address and port). When you use persistence, the BIG/ip Controller sends a series of related connections received from the same client to the same server for processing. The set of related connections is referred to as a ***persistent session***.

For example, say a client wants to purchase an airline ticket from a web site, but during the first connection to the site, the client only reserves the ticket. In order to complete the transaction and purchase the reserved ticket, the client must reconnect to the web site and continue the transaction. When the client returns to the web site to purchase the reserved ticket, the BIG/ip Controller recognizes the client's connection as belonging to the persistent session. Instead of load balancing the connection to a random server, the BIG/ip Controller connects the client to the server that originally processed the reservation. Now the client can complete the transaction using the information stored on the that server.

Note that whether clients need to reconnect to a specific server depends on how the site content is stored and managed. Sites that use back-end database server to manage sales transactions may not necessarily need to use persistence.

The BIG/ip platform allows you to configure persistence behavior for TCP, UDP, and SSL persistence.

- You can set the timeout for a persistent session to start at the beginning of the first connection in the session and run until the timeout expires. Or, you can set the persistence timeout to start each time a packet is received on a persistent connection; this effectively prevents the persistence timer from counting down as long as there is network traffic.
- You can set the BIG/ip Controller to maintain persistence across all virtual servers. This allows all persistent connections coming from the same client to be directed to the same server, regardless of which virtual servers the client is connecting to.
- You can set the BIG/ip Controller to maintain persistence on a particular virtual address. This allows all persistent connections to the same virtual address from a client to be sent to the same server, regardless of which virtual server the client is connecting to.

Working with configuration and monitoring tools

The BIG/ip platform offers a variety of administrative tools, and also supports useful management-oriented protocols to help create a robust and secure administrative environment. For example, the BIG/ip platform supports SNMP and SMTP (outbound only) protocols, that you can use for performance monitoring and notification of system events. The BIG/ip platform also includes useful administrative applications and utilities, such as the BIG/config web application, and the F-Secure SSH client, which provides a secure UNIX shell connection to the BIG/ip Controller from a remote workstation.

Administrative tools

The BIG/ip platform provides three basic tools that you can use to configure and monitor a BIG/ip Controller:

- **The First-Time Boot utility**
  The First-Time Boot utility walks you through initial installation tasks, including defining a root password and setting the external interface. It also walks you through configuration of the BIG/ip web server. The BIG/ip web server hosts the BIG/config application, and also provides convenient downloads such as the F-Secure SSH client, and the SNMP MIB.
- **The BIG/config web application**
  BIG/config is a web application that allows you to both configure and monitor the BIG/ip system. In the BIG/config application, you can configure virtual servers, define IP and packet rate filters, and also configure system objects including the SNMP daemon and system settings. The BIG/config application allows you to monitor performance of several items including virtual servers, IP packet and rate filters, and the BIG/ip system itself. Note that BIG/config requires Netscape Navigator or Microsoft Internet Explorer, versions 4.0 or higher.
- **The BIG/pipe and BIG/top command line utilities**
  The BIG/pipe command line utility allows you to configure and monitor all aspects of the BIG/ip Controller. The BIG/top utility

provide real-time system monitoring. You can use either of these command line utilities directly on the BIG/ip Controller, or from a remote workstation (when connected with the F-Secure SSH client Telnet).

Security features

The BIG/ip platform provides important security features including:

- Data encryption
- Password authentication
- Timeout for inactive connections
- Strict control over which ports are accessible

The BIG/ip Controller provides SSL security (US products only) for the BIG/config application. It also allows you to set IP addresses from which administrative commands and requests are accepted. Password authentication ensures security and serves as a check point for the network. For use with command line utilities, the BIG/ip platforms includes a commercial version of SSH (*F-Secure* from Data Fellows), which guarantees secure, remote access via encrypted sessions.

The BIG/ip Controller can also work in conjunction with other supplementary security products that you may use in your network environment.

What's new in version 2.0

The BIG/ip platform offers major new features in version 2.0, such as BIG/config, the administrative web application, simple configuration of IP filters and rate classes, and Extended Application Verification. The following sections highlight some of the new or enhanced features included in the 2.0 version of the BIG/ip platform.

New terminology

The BIG/ip 2.0 platform incorporates some important terminology changes to help describe the product features and elements more accurately. These changes are reflected in the documentation, and also in the BIG/config application. However, the original commands in the BIG/pipe command line utility are unaffected by terminology changes.

The terminology changes include the following:

- The term "VIP" is replaced by ***virtual server***, and it is used to refer to a specific combination of a virtual server address and a virtual port number. In previous documentation, "vip" was somewhat ambiguous, and was used to refer to a virtual address, or to a specific combination of a virtual address and a virtual port.
- The term ***node*** refers to a specific combination of a node address and a node port. In previous documentation, "node" was often used to refer to a server, rather than a specific port on a server.
- A ***virtual server mapping*** is the list of one or more nodes to which a virtual server has a path.
- The "BIG/ip^2" and "BIG/ip^3" notation is now obsolete. The BIG/ip^2 hardware configuration is now named BIG/ip HA, and the BIG/ip^3 hardware configuration is now named BIG/ip HA+. In addition, the term ***BIG/ip redundant system*** is used to refer to two BIG/ip Controller HA or HA+ units configured for fail-over. An additional model named BIG/ip LB is also available, which includes only a single BIG/ip Controller unit that supports a limited feature set.
- In BIG/ip redundant systems, the terms "master" and "slave" are replaced by ***active*** and ***standby***. The active BIG/ip Controller is that which processes connections, and the standby BIG/ip Controller is that which takes over should the active BIG/ip Controller go down and initiate a fail-over.
- The term "service ping" is replaced by ***service check***. Service check is more appropriate because the action itself involves connecting to a port and verifying that a service is up and running; it does not make use of the standard echo pings that the BIG/ip Controller uses for node ping.

- The term "active service ping" is replaced by two terms: ***ECV service check*** and ***EAV service check***. ECV service check uses ***Extended Content Verification***, which determines whether a node is available based on a send string and receive string specified by the user. EAV service check uses ***Extended Application Verification***, and essentially performs the same type of function as ECV service check. However, EAV service check relies on an external program, often developed by the customer, to perform the actual service check and verify that specific site content is available.

BIG/config and the see/IT application suite

An important addition to the BIG/ip platform in version 2.0 is the BIG/config application. In BIG/config, you can configure and monitor virtually all aspects of the BIG/ip Controller in a user-friendly environment. BIG/config is a component of the see/IT™ application suite, which offers advanced tools for configuration and monitoring for both the BIG/ip Controller and the 3DNS Controller. see/IT also provides statistical analysis of historical data, allowing you to make important site management decisions based on known trends and system behavior. For more information about working with BIG/config, refer to Chapter 4.

Extended Content Verification and Extended Application Verification

Extended Content Verification (ECV) and Extended Application Verification (EAV) allow the BIG/ip Controller to determine whether a node is *up* or *down* by checking to see if specific site content is available. Similar to simple node ping, the BIG/ip Controller performs this check at user-defined intervals. To verify content, ECV uses simple regular expressions that you can define in BIG/config or on the command line, but EAV uses custom applications, which can be provided by the customer, or by the customer in conjunction with F5 Labs. For information about working with both of these features, see Chapter 7.

Simple configuration for IP filters and rate filters

The BIG/config application allows for easy configuration of simple IP filters and rate filters. You can configure filters that allow or deny traffic going to specific virtual servers, or traffic going out to specific sites on the external network or the Internet. The BIG/ip Controller itself supports any IP filter or rate filter definition allowed by BSD/OS IP filtering.

New load balancing features

The BIG/ip platform offers three important new load balancing features:

- Load balancing across arrays of routers and router-like devices
- Enhanced SSL persistence
- Load balancing across groups of servers using priority levels

Load balancing for routers and router-like devices

The BIG/ip platform now supports a new mode, ***Transparent Node Mode***, in which the BIG/ip Controller performs load balancing for routers and router-like devices, such as transparent firewalls or cache servers. There are special planning and configuration issues that you need to address if you want to make use of this feature. Refer to Chapter 7 for more information.

Enhanced support for SSL persistence

The BIG/ip Controller provides enhanced support for SSL persistence. You can configure SSL persistence on individual virtual servers. The BIG/ip platform now includes a system control variable that allows you to change the persistence timer itself. You can set the timer to start when a new persistent connection is established, or you can set the timer to start when the most recent session in a series of persistent connections is established. For information about SSL persistence, refer to Chapter 9.

Priority mode

The BIG/ip platform includes another static load balancing mode, called ***Priority*** mode. In Priority mode, you assign each server to a group, and each group of servers has a priority level. The BIG/ip Controller performs round robin connection distribution to the servers in the highest priority group until the servers in that group become unavailable. Once all the servers in the highest priority group are unavailable, the BIG/ip Controller begins distributing connections to servers in the next lower priority group. For more information about working with Priority mode, see Chapter 9.

The BIG/ip SNMP MIB

The BIG/ip platform includes an SNMP MIB, which exposes statistical information for elements such as virtual server traffic and node performance. The SNMP MIB is compatible with standard SNMP management packages, and you can easily configure SNMP settings in the BIG/config application. For more information about working with the SNMP MIB, refer to Chapter 7.

Optimization for large configurations

You can now set special properties that help optimize performance for large configurations (configurations in excess of 1,000 virtual servers or nodes). For example, you can reduce the number of node pings issued by the BIG/ip Controller, and you can also reduce the amount of ARP traffic that may pass through your network. For more information about optimizing large configurations, see Chapter 7.

New BIG/pipe commands and system control variables

The BIG/pipe command line utility offers new commands, as well as new parameters for existing commands. BIG/pipe also supports a new curly bracket syntax for configuration files. For more information, refer to the *BIG/pipe Command Reference* in Appendix B.

There are new system control variables available in the 2.0 version of the BIG/ip platform, and the default settings for certain existing system control variables are changed from prior versions. To view a description of the system control variables that affect BIG/ip features, and to view their default settings, see Appendix C.

2

Preparing for Installation

- **Planning an installation**
- **Understanding virtual servers**
- **Preparing network components**
- **Preparing site content**
- **Gathering important configuration information**

Planning an installation

This chapter provides detailed information about configuration planning issues that you need to address before installing and configuring the BIG/ip Controller. The chapter outlines how virtual servers work, and explains the type of information you need to prepare before you define virtual servers on the BIG/ip Controller. It also covers other important issues such as how to configure network routing, and how to set up and distribute site content before you actually connect the BIG/ip Controller to the network.

Planning standard configurations

Planning a standard configuration includes the following tasks:

- Draw a topology of your virtual servers.
- Evaluate whether your site content is properly distributed among your servers.
- Verify that the existing network is configured properly.
- Collect information that you need during configuration, such as port numbers, IP aliases, subnet masks, and IP addresses for routers, name servers, virtual servers, and network address translations.

Planning advanced configurations

The BIG/ip Controller supports the following features for advanced configurations which require additional planning and implementation outside the standard configuration:

- Large configurations that manage thousands of virtual servers or thousands of nodes.
- Extended Content Verification for nodes on which you use ECV service check to verify availability of site content.
- Extended Application Verification for nodes where you want to use a custom program to perform an ECV service check.
- Special virtual servers that manage and load balance connections specifically for transparent network devices such as transparent firewalls.

For information about planning and implementing advanced configurations, refer to Chapter 7.

Understanding virtual servers

Each content site that a BIG/ip Controller manages has a virtual server associated with it. A ***virtual server*** is a specific combination of a virtual address and virtual port. The ***virtual address*** is that which is advertised to clients, and it should be the same IP address that is registered for the site's host and domain name. The ***virtual port*** should be the same TCP or UDP port number that is known to client programs. For example, the F5 Labs web site *www.f5.com* resolves to a specific virtual address, and the F5 Labs web server is accessed through virtual port 80 (the standard HTTP port); thus, the F5 Labs virtual server is identified as **`www.f5.com:80`**.

You can control several attributes of virtual servers, virtual addresses, and virtual ports. Note that a virtual address may host one or more virtual servers (see Figure 2.1 on page 2- 2-4 for an example). Also note that virtual servers have a default netmask and a broadcast address. The BIG/ip platform allows you to override a virtual server's default netmask and broadcast address with a custom netmask and broadcast address, which is useful for administrators who manage complex network configurations.

Each virtual server maps to at least one physical port on a physical server, referred to as a ***node***. A virtual server typically maps to several different nodes, as seen in the example in Figure 2.1. The BIG/ip Controller uses a load balancing mode to determine how individual site connections should be distributed among the nodes to which a virtual server is mapped.

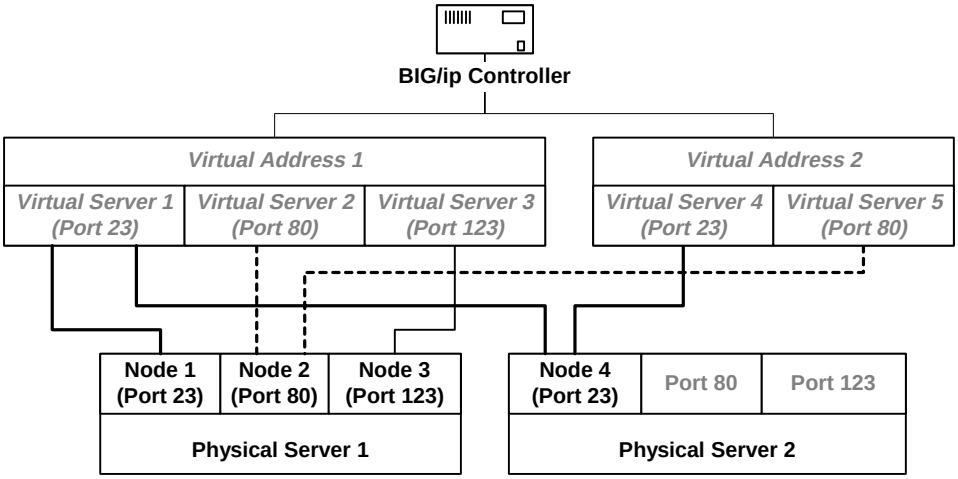

Figure 2.1 *Virtual server mappings*

In Figure 2.1, a BIG/ip Controller manages two virtual addresses, each of which hosts multiple virtual servers. Virtual Address 1 hosts three different virtual servers, one on port 23, one on port 80, and one on port 123. Virtual Address 2 hosts two virtual servers, one on port 23, and the other on port 80.

Each of these virtual servers maps to a node. For example, Virtual Server 1 maps to Node 1 on Physical Server 1, and also maps to Node 4 on Physical Server 2. Some of the nodes shown in Figure 2.1 support more than one virtual server. For example, Node 2 supports Virtual Server 2, and also supports Virtual Server 5.

There are four essential components included in any virtual server mapping:

- A virtual IP address or domain name
- A virtual port number or service name
- Node IP addresses or host name
- Node port numbers or service name

Note that virtual server mappings typically include multiple nodes, and each node included in the mapping is referred to as a ***member*** of the virtual server. Nodes are often members of more than one virtual server.

Figure 2.2 shows an alternate node implementation, which some administrators may find useful in solving bandwidth problems on specific content server platforms. You can create multiple nodes on the same physical server to handle the same Internet service.

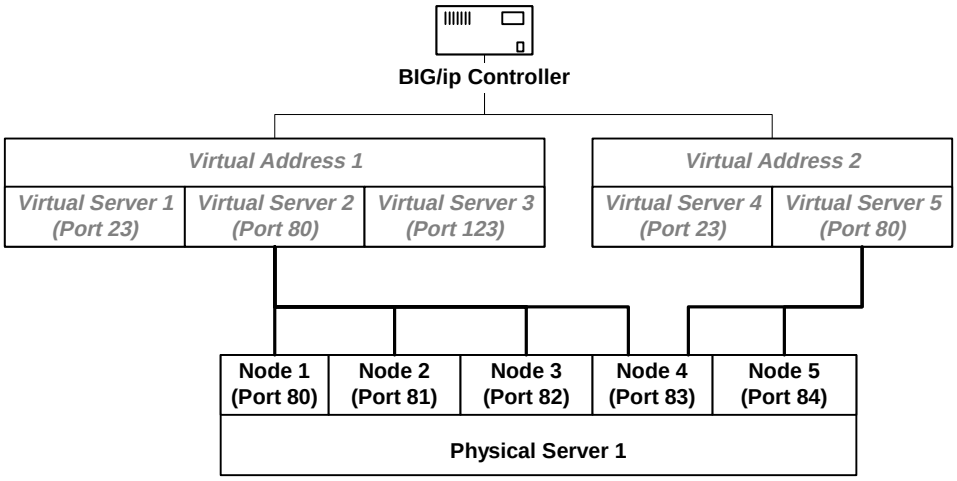

Figure 2.2 *Distributing an Internet service across multiple ports*

In Figure 2.2, Virtual Server 2 accepts connections for HTTP services, which are load balanced across four different ports on Physical Server 1: 80, 81, 82, 83, and 84. The BIG/ip Controller also load balances all HTTP connections received by Virtual Server 5 across ports 83 and 84 on Physical Server 1. Note that both virtual servers are mapped to port 83 on the physical server.

A dynamic load balancing mode works well in this type of configuration. A dynamic load balancing mode allows the BIG/ip Controller to continuously monitor the performance of each node,

and it distributes connections so that nodes used by more than one virtual server, such as Node 4 shown in Figure 2.2, do not become overloaded.

Note

The BIG/ip platform also supports special virtual servers that load balance connections for other types of network devices, such as transparent firewalls, transparent cache servers, routers, and other router-like devices. For more information about these more complex configurations, refer to Chapter 7.

Property settings for virtual servers

There are three sets of properties that apply to virtual servers:

- Virtual server properties apply to a specific virtual server.
- Virtual address properties apply to all virtual servers that have the same virtual address.
- Virtual port properties apply to all virtual servers that include the specific virtual port number or service name.

Property settings for virtual servers

Once you define a virtual server, you can set properties on the virtual server. For example, you can set a connection limit for the virtual server, and you can configure persistence settings for SSL connections. You can also enable or disable a virtual server. The enable/disable feature allows you to take a virtual server down for maintenance without interrupting any of the virtual servers' current connections. When you disable a virtual server, it does not accept new connections, but it allows the current connections to complete, before shutting down completely.

Property settings for virtual addresses

The BIG/ip Controller allows you to configure basic properties for a virtual address including a connection limit, and a netmask and broadcast address. The default netmask is 255.255.255.0, and the

default broadcast address is a combination of the virtual address and the netmask. You can override the default netmask and broadcast address if necessary.

All virtual servers that have the same virtual address inherit the properties of the virtual address.

Property settings for virtual ports

For convenience, the BIG/ip Controller allows you to define default configuration settings for a virtual port number or service name. Each virtual server that uses the port number or service name inherits the default properties for that port number or service. The only default property setting that a specific virtual server can override is whether the port is enabled or disabled for that virtual server.

The configurable settings for a virtual port include:

- Whether the port is currently enabled or disabled.
- A connection limit.
- A time-out for idle connections.
- Persistence settings for TCP and UDP sessions.

Configuration settings for nodes

There are also three sets of properties that apply to nodes:

- Node properties apply to a specific node.
- Node address properties apply to all nodes that are hosted by the node address.
- Node port properties apply to all nodes that include the specific port number or service name.

Property settings for nodes

Once you define a node, you can set specific properties on the node itself including a connection limit, and special content verification settings. You can enable or disable a node, which makes the node available, or unavailable, to accept new connections. If you disable a node while it is currently hosting connections, the node allows

those connections to complete, but does not allow any new connections to start. This is useful when you want to take a node down for maintenance without interrupting network traffic.

Property settings for node addresses

Node addresses have property settings that apply to all nodes hosted by the node address. Node address property settings include:

- Whether the node address is currently enabled or disabled
- A connection limit
- A load balancing ratio weight or priority level used when the load balancing mode is set to Ratio or Priority
- An IP alias that the BIG/ip Controller can ping instead of the true node address

Aliases for node addresses are useful for BIG/ip Controllers than manage thousands of nodes. For more information about optimizing large configurations, see Chapter 7.

Property settings for node ports

You can set global properties for port numbers or service names used by nodes. These settings apply to all nodes that include the port number or service name, regardless of which physical server hosts the node. You can override all global node port properties for specific node except the service check frequency and service check timeout settings. Node port properties include:

- Whether the node port is currently enabled or disabled.
- A service check frequency and timeout.
- ECV service check settings, including content strings and receive rules.

Using advanced service check options

If you plan on using advanced service check options such as extended content verification or extended application verification, you should review the corresponding sections of Chapter 7. Extended content verification requires that you specify send and receive strings, which are defined as regular expressions. Extended

application verification requires a custom-developed program that performs the actual service check on behalf of the BIG/ip Controller. Both of these options require additional planning and configuration issues.

Preparing network components

Before you install a BIG/ip Controller in your network, you need to make sure that your network meets several requirements. The existing network should be fully functional, and it should support one or more IP services. Several individual network components must also meet specific requirements including routers, hubs, gateways and content servers.

Router configurations

The BIG/ip Controller must communicate properly with both the network router and the content servers that the BIG/ip Controller manages. Because there are a multitude of router configurations and varying levels of direct control an administrator has over each router, you need to carefully review the router configurations in your own network, and evaluate whether you need to change any existing configuration before you install the BIG/ip Controller.

Each router connected to the BIG/ip Controller must be IP compatible, and the router's interface must be compatible with the external interface on the BIG/ip Controller (either IEEE 802.3z/Ethernet or FDDI, depending on the model of BIG/ip Controller that you purchase).

- The default route for the BIG/ip Controller must be set to the gateway address of the router connected to the BIG/ip Controller's external interface (the interface from which it receives connection requests). You can set the default route during the First-Time Boot configuration, or you can set the default route by editing the */etc/netstart* file.

- The routers connected to the BIG/ip Controller's external interface must have appropriate routes to get to all of the virtual addresses hosted by the BIG/ip Controller, and to get to the BIG/ip Controller's administrative address.

Routing between a BIG/ip Controller and a router

The BIG/ip Controller is designed to eliminate the need for administrators to modify routing tables on a router that routes to a BIG/ip Controller. The BIG/ip Controller uses Address Resolution Protocol (ARP) to notify a router of the IP addresses of its external interface as well as its virtual servers. The BIG/ip Controller supports static route configurations, dynamic routing (via BGP4, RIP1, RIP2, and OSPF), and subnetting.

You may use dynamic routing with the BIG/ip Controller, but it is not normally required. Refer to Chapter 6 for information about implementing dynamic routing in a BIG/ip system environment.

Routing between a BIG/ip Controller and content servers

All network traffic coming into and going out of the content servers in the array must pass through the BIG/ip Controller. In order for routing to these servers to work properly, you need to set each server's default route to be the IP address of the BIG/ip Controller internal interface.

Content servers

All content servers managed by the BIG/ip Controller must have TCP/IP-compliant operating systems. For each server included in the server array, you should verify the following information and have it available when you begin configuring the BIG/ip Controller:

- Verify that the ports on the content server are properly configured for the Internet services that the content server needs to support.
- Verify that each server has at least one unique IP address defined. Note that a BIG/ip Controller can use multiple IP aliases defined on the content server as node addresses.
- Verify that the content server is communicating with other devices on the network.

Each TCP/IP service supported by the BIG/ip virtual servers must be configured on at least one of the content servers in the array. For specific information about configuring TCP/IP servers, and verifying TPC/IP services on specific ports, refer to the documentation provided by the server manufacturer.

Setting up content servers on different logical networks

A content server can be installed on a different logical network than that of the BIG/ip Controller, as long as the path of the content server's default route goes through the BIG/ip Controller. If your network environment includes this type of configuration, you need to modify the */etc/rc.local* file on the BIG/ip Controller. The */etc/rc.local* file stores the BIG/ip Controller's routing information, and you can edit it in a UNIX editor, such as vi or pico.

With this type of network configuration, you need to resolve one of two different routing issues, depending on whether the logical networks are running on the same LAN.

If the logical networks are on the same LAN, they either share media directly, or they have a switch or a hub between them. In this configuration, you need to add an interface route to the BIG/ip Controller's internal interface. For example, if the BIG/ip Controller's internal interface were on logical network 192.168.5/24, and a content server's were on logical network 192.168.6/24, you would need to add the following line to the */etc/rc.local* file:

```
route add -net 192.168.6 -interface exp1
```

If the logical networks are on different LANs, they have a router between them. In this environment, you need to do three things:

- On the BIG/ip Controller, you need to add a static gateway route to the */etc/rc.local* file. In the example above, where the BIG/ip Controller is on logical network 192.168.5/24 and the content servers are on logical network 192.168.6/24, you would need to add the following line to the */etc/rc.local* file:

```
route add -net 192.168.6.0 -gateway \
  192.168.5.254
```

- On each content server, you need to set the default route to point to the router between the LANs. The content server's default route using the above example would be:
  `route add default -gateway 192.168.6.254`
- On the router between the LANs, you need to set the default route to the internal interface address on the BIG/ip Controller. The router's default route using the above example would be:
  `route add default -gateway 192.168.5.200`

Administrative workstations

You can access a BIG/ip Controller from a remote workstation in two ways:

- The BIG/config application is a web-based application that runs in a browser. BIG/config connects to a port on the BIG/ip Controller and supports two security options. You can set a login password , and you can also define an IP address, or a range of IP addresses, from which the BIG/config application accepts connections.
- The F-Secure SSH data encryption client (included with the BIG/ip platform) allows you direct access to the system from a remote workstation over an SSH connection. From the F-Secure SSH client, you can use BIG/ip command line utilities, such as BIG/pipe and BIG/top, as well as basic UNIX system commands.

Some administrators may find it convenient to use both of these administrative options. Depending on your site requirements, you may also want to take advantage of SNMP on the BIG/ip Controller. You can configure SNMP from the BIG/config application, and then use it with SNMP management systems of your own.

Note

You can also locally configure and manage a BIG/ip Controller via a VGA monitor and keyboard connected directly to the unit.

Preparing site content

Site content for each virtual server that the BIG/ip Controller manages can be configured in one of two ways:

- Content can be locally stored on the servers in the array, and accessed directly on the servers. This is typical of static site content, which is not modified by clients.
- Content can be distributed on one or more file servers and accessed via the servers in the server array. This is typical of content that is modified by clients, such as the items in a shopping cart on an e-commerce site.

Static web site content

If your web site content is read-only, you can use a distributed, replicated content scheme. With a replicated content scheme, the content on one server is identical to that of the other servers managing content for the same web site. This ensures that all client requests access the same content, no matter which physical server they are actually connected to.

Stateful site content

If your site content is dynamic, such as that created with Active Server Pages, we recommend that you store the stateful information, if not all the content, on a single shared file server. This allows the BIG/ip Controller to continue to use a load balancing algorithm to control traffic across the server array. For best performance, the shared file server should be situated in the array that is managed by the BIG/ip Controller.

If you maintain stateful site content on individual servers instead of a shared file server or back-end database, you need to configure TCP or SSL persistence on the virtual servers managed by the BIG/ip Controller. TCP and SSL persistence allow clients to disconnect from a site, and later reconnect to the site and continue a previous session. If TCP or SSL persistence is enabled, the BIG/ip Controller connects the client to the node that hosted the client's

original session (as long as the persistence timeout has not expired). For more information about persistence settings, refer to *Working with persistence* in Chapter 9.

Gathering important configuration information

When you are planning an installation, you may find it helpful to create a hierarchical view of the virtual server mappings that you need to define during the configuration process. This is particularly useful if you are designing a large or complex configuration. Also note that the virtual servers you define on the BIG/ip Controller become an integral part of your network. We recommend that you add the virtual servers and corresponding mapping information to your overall network documentation.

Before you begin the installation and configuration process, you should also to determine appropriate property settings for nodes, node addresses, and node ports. This is especially important if you want to take advantage of the BIG/ip platform's extended content verification feature, which allows you to connect to a node and verify that specific site content is accessible. Other property settings that you need to define should be based on the speed and processing ability of each server in the array, as well as the type of traffic you expect the node to handle.

3

Installation and Initial Configuration

- **Unpacking and installing the hardware**
- **Configuring the BIG/ip system**
- **Preparing to configure BIG/ip software**

Unpacking and installing the hardware

The following checklists outline both the hardware provided with your BIG/ip redundant system, as well as the peripheral hardware that you must supply.

Equipment provided with a BIG/ip redundant system

For each BIG/ip Controller in the system, F5 Labs provides you with the following items:

- One power cable
- One PC/AT-to-PS/2 keyboard adapter
- Four rack mounting screws
- Two keys for the front panel lock
- One extra fan filter

In addition, F5 Labs provides you with:

- One fail-over cable (to connect the two units in the redundant system together)
- One *BIG/ip Installation and User's Guide*
- One *F-Secure SSH User's Guide* (USA products only)

Note

Additional documentation, including technical notes and frequently asked questions, is available in the Technical Support section of F5 Labs' web site at ***http://tech.F5.com****. To access this site, you need to obtain a customer ID and a password from your F5 service engineer.*

Peripheral hardware that you provide

For each BIG/ip Controller in the system, you need to provide the following peripheral hardware:

- Either a VGA monitor and PC/AT-compatible keyboard, or a serial terminal and a null modem cable, for direct administrative access to the BIG/ip Controller.

- Two network devices that are compatible with the network interface cards installed on your BIG/ip Controllers. The devices can support either 10/100 Ethernet or FDDI/CDDI (including multiple FDDI and full duplex).
  - For Ethernet you need either a 10Mb/sec or 100 Mb/sec hub or switch.
  - For FDDI/CDDI you need either a concentrator or a switch.

We also recommend that you have a remote administrative workstation in place from which you can configure and monitor each BIG/ip Controller in the redundant system.

Familiarizing yourself with the BIG/ip Controller hardware

Before you being to install the BIG/ip redundant system, review the figures below which illustrate all controls and ports on both the front and the back of a BIG/ip Controller unit. The hardware installation instructions refer to ports and other controls on the unit by the numbers they are identified by in the following figures.

Figure 3.1 shows the front of a BIG/ip Controller, where you can turn the unit on or you can reset the unit.

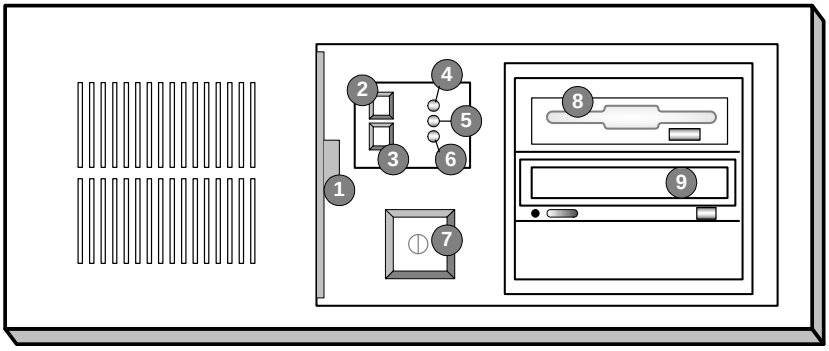

Figure 3.1 *Front view of a BIG/ip Controller*

1. Fan filter

2. Keyboard lock
3. Reset button
4. Keyboard lock LED
5. Hard disk drive LED
6. Power LED
7. On/off button
8. 3.5 floppy disk drive
9. CD-ROM drive

Figure 3.2 shows the back of a BIG/ip Controller. Note that all ports are labeled, even those which are not intended to be used with the BIG/ip Controller.

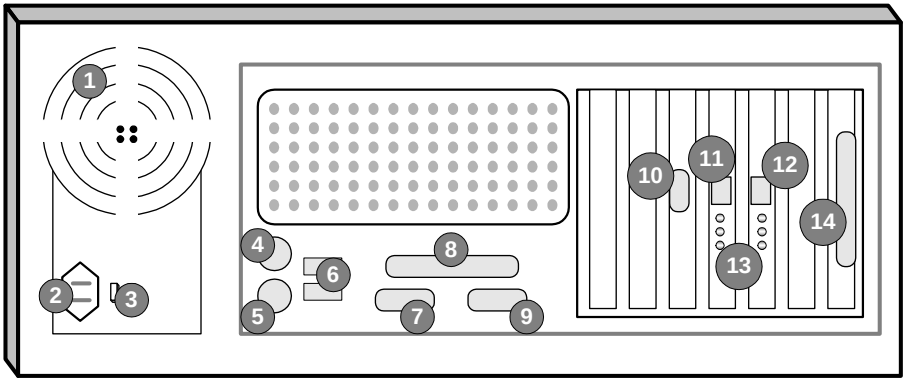

Figure 3.2 *Back view of a BIG/ip Controller*

1. Fan
2. Power in
3. Voltage selector
4. Mouse port*
5. Keyboard port
6. Universal serial bus ports*

7. Terminal serial port
8. Printer port*
9. Fail-over port
10. Video (VGA) port
11. Internal interface (RJ-45)
12. External interface (RJ-45)
13. Interface indicator LEDs
14. Watchdog card*

Note

**Ports marked with an asterisk (*) are not used by the BIG/ip Controller, and do not need to be connected to any peripheral hardware.*

Environmental requirements and usage guidelines

A BIG/ip Controller is an industrial network appliance, designed to be mounted in a standard 19 inch rack. To ensure safe installation and operation of the unit, be sure to take the following into consideration before you install the unit in the rack:

- The rack itself should be installed according to the manufacturer's instructions, and should be checked for stability before you install any BIG/ip Controller hardware.
- The maximum air temperature in the room can not exceed 50° C. Internal temperatures should be considered for continued safe operation.
- The rack should be structured and positioned so that once the BIG/ip Controller is installed the power supply and the vents on both the front and back of the unit are unobstructed. There should be adequate ventilation around the unit at all times.
- The branch circuit into which you plug the unit should not be shared by more electronic equipment than it is designed to manage at one time.

- The unit must be connected to Earth ground, and it should have a reliable ground path maintained at all times.

WARNING

The BIG/ip Controller contains a lithium battery. There is danger of an explosion if you replace the lithium battery incorrectly. We recommend that you replace the battery with only the same or an equivalent type of battery originally installed in the unit. Be sure to discard all used batteries according to the manufacturer's instructions.

Installing and connecting the hardware

There are six basic steps to installing the hardware. Note that you should not turn a BIG/ip Controller on until all of the peripheral hardware is connected to the unit.

To install the hardware

1. Insert the BIG/ip Controllers in the rack and secure each using the four rack mounting screws provided.
2. Connect the hardware that you have chosen to use for input/output:
   - If you are using a VGA monitor and keyboard, connect the monitor connector cable to port number 10 and the keyboard connector cable to port number 5, as shown in Figure 3.2, on page 3-4. Note that a PC/AT-to-PS/2 keyboard adapter is included with each BIG/ip Controller (see the packing list on page 3-2).
   - If you are using a serial terminal, connect the null modem cable to port 7, as shown in Figure 3.2. Configure the serial terminal settings as follows:

     -9600 baud

     -8 bits

     -1 stop bit

     -No parity

3. Connect the external interface (port 12 in Figure 3.2) to the network from which the BIG/ip Controller receives connection requests. In a normal configuration, this is typically the network connected directly to the Internet or other external network. In a Transparent Node Mode configuration, this is typically your internal network.
4. Connect the internal interface (port 11 in Figure 3.2) to the network that houses the array of servers, routers, or firewalls that the BIG/ip Controller load balances. In a normal configuration, this is typically the internal network that houses your content servers. In a Transparent Node Mode configuration, this is typically the external network that houses your routers, transparent firewalls, or other transparent devices.
5. Connect the fail-over cable to port number 7 on each unit as shown in Figure 3.2.
6. Connect the power cable to the BIG/ip Controller, and then connect it to the power source.

Working with more than two NICs

The BIG/ip Controller is available with more than two network interface cards (NICs). If you have purchased a unit with three or more NICs, be sure to note down how you connect the cables to the internal and external interfaces. The First Time Boot Utility automatically detects the number of interfaces that are installed and prompts you to configure more external interfaces, if you wish. It's important to select the correct external interface based on the way you have connected the cables to the back of the unit.

Once you complete these steps, you are ready to define system elements that allow you to access the BIG/ip Controller from the network. You perform this task using the First-Time Boot utility, which runs automatically when you start the BIG/ip Controller for the first time. Note that the First-Time Boot utility prompts you to enter specific configuration information. We do not recommend that you turn the unit on until you have compiled the necessary information and are ready to enter it in the system.

Configuring the BIG/ip system

The first step in configuring a BIG/ip Controller is to run the First-Time Boot utility. This utility walks you through a brief series of required configuration tasks, such as defining a root password, and configuring external and internal interfaces to the system. Until you complete this process, you cannot access the unit from the network.

The First-Time Boot utility creates the following files, which store basic BIG/ip system configuration settings:

- An administrative IP access file
- An interfaces table
- The */etc/bigip.conf* file
- The */etc/netstart* file
- The */etc/hosts* file
- The */etc/ethers* file

The First-Time Boot utility allows you to review and confirm configuration settings before it saves them and completes the configuration process.

The First-Time Boot utility also prompts you to configure the BIG/ip web server, which hosts the BIG/config application. For example, you need to define a user name and password, and you also need to provide an IP address from which access is permitted. Until you complete this process, you cannot access the BIG/config application, nor can you access convenient downloads, such as the F-Secure SSH client and the SNMP MIB.

Once both configuration utilities run, the primary system configuration is complete. Before you continue with the BIG/ip Controller configuration, you may want to define host names for network devices, virtual servers, and nodes in the */etc/hosts* file.

Booting the BIG/ip Controller and running the First-Time Boot utility

To boot the BIG/ip Controller, turn on the power switch located on the front of the BIG/ip Controller chassis (see Figure 3.1, number 7). Note that some models may have the power switch on the back. When the BIG/ip Controller is successfully powered up, you see the Welcome screen shown in Sample Screen 3.1.

```
W E L C O M E

              Thank you for purchasing BIG/ip(tm)!
                 the BIG/ip Controller - 2.0.1
             Copyright (C) 1996-1999, F5 Labs, Inc.
                      All Rights Reserved
                    For Technical Assistance:
                    Voice: +1 (206) 505-0800
                     Email: support@f5.com

                 [ press any key to continue ]
```

Sample Screen 3.1 *Initial BIG/ip Controller screen*

To start the First-Time Boot utility from this screen, simply press any key on the keyboard. Once the First-Time Boot utility begins to run, the screen shown in Sample Screen 3.2 opens.

```
                        First-Time Boot
                  System Configuration Utility

Welcome to BIG/ip(tm).   Before using your
BIG/ip(tm), you will have to configure the
root password, BIG/ip(tm) host name, and
interface cards.
This utility will take you through this
process step-by-step.

Before any configuration files are written to
disk, you will be asked to confirm all your
selections.

       [ Press ctrl-E to exit and configure manually ]

                [ press any key to continue ]
```

Sample Screen 3.2 *System Configuration Utility*

Once you press a key to continue the process, the First-Time Boot utility prompts you for the following information, in order:

- Root password
- Host name
- Interface settings for the external network interface
- Interface settings for the internal network interface
- Configuration for BIG/ip redundant systems
- IP address for remote administration
- Default route (typically a router's IP address)

Defining a root password

A root password allows you administrative access to the BIG/ip Controller system. The password must contain a minimum of 6 characters, but no more than 128 characters. Passwords are case sensitive, and we recommend that your password contains a

combination of upper and lowercase, as well as punctuation characters. Once you enter a password, the First-Time Boot utility prompts you to confirm the root password.

WARNING

Once you define and confirm the root password at this screen, you cannot change the root password until the First-Time Boot utility completes and you reboot the BIG/ip Controller (see Chapter 6). Note that you can change other system settings when the First-Time Boot Configuration utility prompts you to confirm your configuration settings.

Defining a host name

The host name identifies the BIG/ip Controller itself. There are no restrictions on host names, other than those imposed by your own network configuration.

Configuring the interface to the external network

When you configure the interface that connects the BIG/ip Controller to the external network, the configuration utility prompts you for the following information:

- Interface IP address
- Netmask
- Broadcast address
- Interface media type

Understand that the IP address of the external network interface is not the IP address of your site or sites. You use the external network interface IP address for remote administration of the

BIG/ip Controller. The IP address of the sites themselves are specified by the virtual IP addresses associated with each virtual server you configure.

Note

The configuration utility lists only the network interface devices that it detects during boot up. If the utility lists only one interface device, the network adapter may have come loose during shipping. Check the LED indicator on the network adapters to ensure that they have detected the available BIG/ip Controller media.

Once you select the appropriate interface, enter the following information:

- **IP address**
- **Netmask**
  Note that the BIG/ip Controller uses a default netmask of 255.255.255.0.
- **Broadcast address**
- **Media type**
  The media type options depend on the network interface card included in your hardware configuration. The BIG/ip platform supports the following types:
  - auto
  - 10baseT
  - 10baseT,FDX
  - 100baseTX
  - 100baseTX,FDX

Configuring the interface to the internal network

When you configure the interface that connects the BIG/ip Controller to the internal network (the servers and other network devices that sit behind the BIG/ip Controller), the First-Time Boot utility prompts you for the following information:

- **IP address**

- **Netmask**
  Note that the BIG/ip Controller uses a default netmask of 255.255.255.0.
- **Broadcast address**
- **Media type**
  The media type options depend on the network interface card included in your hardware configuration. The BIG/ip platform supports the following types:
  - auto
  - 10baseT
  - 10baseT,FDX
  - 100baseTX
  - 100baseTX,FDX

Configuring settings for a BIG/ip redundant system

If you have a BIG/ip redundant system, you need to enter specific configuration information at this point. If you do not have a BIG/ip redundant system, the First-Time Boot utility allows you to go directly to the next step in the configuration process where you define an administrative IP address.

Each unit in a BIG/ip redundant system configuration uses unique internal and external IP addresses. However, in order for connections to be routed to the active BIG/ip Controller in a redundant system, you need to define two IP aliases that will be shared between the two BIG/ip Controllers in the redundant system:

- An external IP alias associated with each unit's external interface
- An internal IP alias associated with each unit's internal interface

The shared IP aliases are actually used only by the active unit in the redundant system. When a fail-over occurs, the IP alias is switched to the newly active machine. All web site connections sent to the BIG/ip Controller from the outside router should be sent to the external IP alias to guarantee that the active machine receives the connection.

Each network device behind the BIG/ip redundant system should have the internal IP alias set as the default route, which again guarantees that the network devices always communicate via the active BIG/ip Controller in the redundant system.

For administration purposes, you can connect to the BIG/ip Controller IP alias, which always connects you to the active machine. To connect to a specific controller, simply connect directly to the external or internal IP address of that BIG/ip Controller.

Configuring the external IP alias

To configure the external IP alias, you need to provide the following information:

- An IP alias
- A netmask
- A broadcast address

Configuring the internal IP alias

To configure the internal IP alias, you need to provide the following information:

- An IP alias
- A netmask
- A broadcast address

Configuring remote administration

In order to provide for remote administration using BIG/config or the F-Secure SSH client, you need to specify a single IP address, or a range of IP addresses, from which administration is allowed. To specify a range of IP addresses, include the asterisk character ("*") as a wildcard character in the IP addresses. The following example allows remote administration from all hosts on the 192.168.2.0 network:

```
192.168.2.*
```

Configure Default Route

If a BIG/ip Controller does not have a predefined static route for network traffic, the unit automatically sends traffic to the IP address that you define as the default route. Typically, a default route is set to a router's IP address.

Configuring settings for the BIG/ip web server

The BIG/ip web server requires you to define a domain name for the server on both the internal and the external interfaces. The BIG/ip web server configuration also requires that you define a user ID and password. On US products, the configuration also generates certificates for authentication.

Note that if you ever change the IP addresses or host names on the BIG/ip Controller interfaces, you need to reconfigure the BIG/ip web server to reflect your new settings. You can run the BIG/ip web server configuration utility from the command line using the following command:

`reconfig-httpd`

If you wish to create a new password for the BIG/ip web server, delete the */var/f5/httpd/basicauth/users* file before running the **`reconfig-httpd`** script. If this file is missing from the configuration, the utility prompts you for both user ID and password information.

You can also add users to the existing password file, change a password for an existing user, or recreate the password file, without actually going through the BIG/ip web server configuration process. For more information, see Chapter 6.

WARNING

If you have modified the BIG/ip web server configuration outside of the configuration utility, be aware that some changes may be lost when you run the **`reconfig-httpd`** *utility. This utility overwrites several BIG/ip web server files, but it does warn you before doing so.*

Confirming configuration settings

The final step in completing the First-Time Boot utility is to confirm your configuration settings. You can confirm or edit the following settings:

- Host name
- Interface settings
- BIG/ip redundant system configuration
- Administrative IP address
- Default route

Once you confirm all of the configuration settings, the configuration utility saves the configuration settings. During this process, the First-Time Boot utility creates the following files and tables:

- An administrative IP access file
- An interfaces table
- A */etc/bigip.conf* file
- A */etc/netstart* file
- A */etc/hosts* file
- A */etc/ethers* file

Defining host names for network devices

Once you complete the First-Time Boot utility, you may want to insert additional host names and IP addresses for network devices into the */etc/hosts* file to allow for more user-friendly system administration. You can define host names for network devices such as routers, network interface cards, and nodes.

The */etc/hosts* file, as created by the First-Time Boot utility, is similar to the following example, shown in Sample Screen 3.3.

```
#bigip host table ( default )
127.0.0.1 localhost localhost.host.domain
# add your default gateway here
207.17.112.254
#  real - external interface
207.17.112.230 bigip ext
# real - internal interface
192.168.1.100 int
# VIPs  ( add as necessary )
# nodes ( add as necessary )
```

Sample Screen 3.3 *The /etc/hosts file created by the First-Time Boot utility*

The sample hosts file lists the IP addresses for the default router, the internal network interface, and the external network interface, and it contains place holders for both the virtual servers and the content servers that your BIG/ip Controller will manage.

Preparing to configure BIG/ip software

Once you complete the First Time Boot Utility, you are ready to configure the BIG/ip software. If you plan on using only the BIG/config web application, you do not need to do any additional installation steps. Refer to Chapter 4 in this manual for information about using the BIG/config application.

If you plan on using command line utilities, such as BIG/pipe and BIG/top, for configuration and system monitoring, you need to set up an administrative workstation with the F-Secure SSH client. The F-Secure SSH client is an interactive shell that provides a secure connection between a remote administrative workstation and the BIG/ip Controller.

Preparing workstations for command line administration

The F-Secure SSH client provides remote access to the BIG/ip system command line from a remote administrative workstation. The BIG/ip First-Time Boot utility automatically configures an F-Secure SSH Server on the BIG/ip Controller, based on the BIG/ip system configuration settings you provide. Your administrative workstation needs the F-Secure SSH client to communicate with the BIG/ip system via the F-Secure Server.

Note

In the First-Time Boot utility, you specify an IP address, or a range of IP addresses, from which remote administration is allowed. You must install the F-Secure SSH client on a workstation which has the IP address, or falls into the range of IP addresses, you specified during the First-Time Boot utility.

The BIG/ip platform includes a version of the F-Secure SSH client for each of the following platforms: Windows, UNIX, and Macintosh. You can download the F-Secure client using your web browser, or you can download the client using an FTP server on the administrative workstation. Note that the F-Secure license agreement allows you to download two copies of the F-Secure SSH client. If you require additional licenses, you need to contact Data Fellows. For information about contacting Data Fellows, as well as information about working with the SSH client, refer to the F-Secure manual included in the BIG/ip product package.

Note

You can also use the F-Secure SSH suite for file transfer to and from the BIG/ip Controller, as well as for remote backups. An F-Secure SSH client is pre-installed on the BIG/ip Controller to assist with file transfer activities. Please refer to the F-Secure User's Manual for more information.

Using a web browser to download the F-Secure SSH client

The F-Secure SSH client is available in the Downloads section of the BIG/ip web server. For US products, you connect to the BIG/ip web server via SSL on port 443 (use **`https://`** rather than **`http://`** in the URL). For international products, you use standard HTTP, unless you have installed SSL on your system. Once you connect to the BIG/ip web server, click the link to Downloads and Documents. From the Downloads page, you can select the SSH Client.

Using an FTP server to download the F-Secure SSH client

You can transfer the F-Secure SSH Client using FTP, as long as the destination workstation has an FTP server installed. After you transfer the installation file, you simply decompress the file and run the F-Secure installation program.

You initiate the transfer from the BIG/ip Controller itself, using the monitor and keyboard, or the serial terminal, attached directly to the BIG/ip Controller.

1. Locate the SSH client appropriate for the operating system that runs on the administrative workstation:

   a) Go to the */usr/contrib/fsecure* directory where the F-secure SSH clients are stored.

   b) List the directory, noting the file name that corresponds to the operating system of your administration workstation.

3. Start FTP:
   `ftp`

4. Open a connection to the remote workstation using the following command, where **`IP address`** is the IP address of the remote workstation itself:
   `open <IP address>`
   Once you connect to the administrative workstation, the FTP server on the administrative workstation prompts you for a password.

5. Enter the appropriate user name and password to complete the connection.
6. Switch to passive FTP mode:
   `passive`
7. Switch the transfer mode to binary:
   `bin`
8. Go to the directory on the administrative workstation where you want to install the F-Secure SSH client.
9. Start the transfer process using the following command, where **`filename`** is the name of the F-Secure file that is specific to the operating system running on the administrative workstation:
   `put <filename>`
10. Once the file is transferred, exit the FTP utility using the following command:
    `quit`

Setting up the F-Secure SSH client on a Windows 95 or Windows NT workstation

The F-Secure SSH client installation file for Windows platforms is compressed in ZIP format. You can use standard ZIP tools, such as PKZip or WinZip to extract the file.

1. Log on to the Windows workstation.
2. Go to the directory to which you transferred the F-Secure installation file. Run PKZip or WinZip to extract the files.
3. The set of files extracted includes a Setup executable. Run the Setup executable and install the client.
4. Start the F-Secure SSH client.
5. In the SSH Client window, go to the File menu and choose Connect.
   The Connect Using Password Authentication window opens.
6. Click Properties.

7. In the Options dialog box, check `Compression` and `Forward X11`, and set the Cipher option to `Blowfish`. Click OK to return to the Connect Using Password Authentication window.
8. In the Connect Using Password Authentication window, type the following items:

   a) BIG/ip Controller IP address or host name

   b) The root user name

   c) The root password
9. Press the Return key to log onto the BIG/ip system.

Setting up the F-Secure SSH client on a UNIX workstation

The F-Secure installation file for UNIX platforms is compressed in TAR/Gzip format.

1. Log on to the workstation and go to the directory into which you transferred the F-Secure SSH client tar file.
2. Untar the file and follow the instructions in the *install* file to build the F-Secure SSH client for your workstation.
3. Start the SSH client.
4. Open a connection to the BIG/ip Controller:

   `ssh -l root [BIG/ip IP address]`
5. Enter the root password.

Configuring and synchronizing BIG/ip redundant systems

You synchronize your units using the BIG/config application or the **`bigpipe configsync`** command. For information on synchronizing with BIG/config, refer to *Synchronizing*

configurations in a redundant system, on page 4-9. For instructions on synchronizing from the command line, refer to *Synchronizing BIG/ip redundant systems*, on page 5-25.

Before synchronizing, you need to make a few changes to your configuration on each unit.

Preparing to synchronize

To use synchronization, you must make the following configuration changes on each BIG/ip Controller:

1. Create a file named */etc/bigip.failover*, containing the real IP address of the internal interface of the other BIG/ip Controller. The file should contain one line in the following format:
   `FailoverIp <ip-addr>`
2. In the */etc/sshd_config* file, verify that the `AllowHosts` line includes the IP address of the other BIG/ip Controller.
3. Run the **`ssh-keygen`** command to generate the */root/.ssh/identity* and */root/.ssh/identity.pub* files that incorporate NULL passphrases. Respond to all questions by pressing the Return key as shown below:

   a) Prompt> ssh-keygen **`<return>`**

   b) Enter file in which to save the key(/root/.ssh/identity): **`<return>`**

   c) Enter passphrase: **`<return>`**

   d) Enter the same passphrase again: **`<return>`**
4. Append the contents of the */root/.ssh/identity.pub* file to the remote BIG/ip Controller's */root/.ssh/authorized_keys* file, using the following command:

```
cat /root/.ssh/identity.pub | ssh -l root \
    <ip-address-of-remote-BIG/ip> 'cat>>
    /root/.ssh/authorized_keys'
```

WARNING

The **`bigpipe configsync`** *command replaces the default configuration file on the second BIG/ip Controller with the current configuration of the BIG/ip Controller from which you execute the command. We recommend that you make backup copies of the configuration files on both systems before executing this command.*

4

Working With the BIG/config Application

- **Using the BIG/config application**
- **Working in the BIG/config window**
- **Setting system properties for the BIG/ip Controller**
- **Configuring virtual servers and nodes**
- **Configuring system redundancy**
- **Configuring IP filters and rate filters**
- **Configuring the BIG/ip SNMP agent**
- **Viewing the Extended Content Verification Summary**
- **Using the BIG/ip System Command for command line access**
- **Viewing system statistics and log files**

Using the BIG/config application

The BIG/config application is a web application that you can use to administrate and monitor the BIG/ip Controller over a secure connection. From the BIG/config application, you can configure a wide variety of items including:

- Virtual servers and nodes
- Redundant system settings
- Network address translations for individual nodes
- IP filters and rate filters
- SNMP settings

You can also monitor various aspects of the system such as real-time performance statistics for virtual servers, nodes, and NATs; security audit information; and system log files.

You can complete all the typical configuration tasks for a BIG/ip Controller system using the BIG/config application. Certain users who want to configure advanced BIG/ip platform features may need to use command line utilities; however, most users should find that the BIG/config application offers access to all of the configuration settings that they need.

Working in the BIG/config window

The BIG/config window is divided into two areas:

- The ***System tree*** provides navigation for the BIG/config application. To open a specific screen, click on the corresponding icon in the System tree. For example, to display a list of virtual servers, click the **Virtual Servers** icon, or, to view IP filter statistics, click **Statistics**, and then select **IP Filters**.
- The BIG/config main window displays the configuration screens, or statistics screens.

Using the System tree

The BIG/config application displays BIG/ip system items in the ***System tree***, shown in the left frame:

- **BIG/ip**
  Displays the BIG/ip System Properties screen where you can configure basic BIG/ip Controller options. From this screen, you can also access the Advanced Properties screen, where you can set BIG/ip system control variables. The icon in the tree also displays the redundancy mode in which the unit is currently running: active or standby.
- **Virtual Servers**
  Displays the Virtual Servers list, which includes all virtual servers managed by the BIG/ip Controller. From this screen you can define new virtual servers, set properties on existing virtual server, virtual addresses, or ports. You can also view the Network Map screen, which provides a hierarchical view of all virtual servers and nodes managed by the BIG/ip Controller.
- **Nodes**
  Displays the Nodes list, which includes all nodes managed by the BIG/ip Controller. From this screen you can set properties on nodes, node addresses, and ports. You can also view the Network Map screen, which provides a hierarchical view of all virtual servers and nodes managed by the BIG/ip Controller.
- **NATs**
  Displays the Network Address Translations list. From this screen, you can define new network address translations for nodes, or you can set properties for existing network address translations.
- **NICs**
  Displays information about the interface cards installed on the BIG/ip Controller. From this screen, you can set fail-over properties for each interface card.
- **IP Filters**
  Displays the list of IP filters running on the BIG/ip Controller. From this screen, you can add new IP filters, or you can change the settings for existing IP filters.

- **Rate Filters**
  Displays the list of rate filters running on the BIG/ip Controller. From this screen, you can add new rate filters, define new rate classes, or you can change the settings for existing rate filters and rate classes.
- **SNMP**
  Displays the BIG/ip Controller SNMP configuration options. In this screen, you can define the options necessary to use the SNMP agent.
- **ECV**
  Displays the Extended Content Verification Summary screen. In this screen, you can view the ECV service check settings for all nodes that use ECV service check. You can also access individual node properties, where you can change the ECV service check settings for the selected node.
- **BIG/pipe**
  Displays the BIG/ip System Command screen where you can execute BIG/pipe commands.
- **Statistics**
  The Statistics icon expands to display icons for all of the statistics screens. In addition to basic system statistics, you can also view statistics on virtual servers, nodes, NATs, IP filters, and rate filters.
- **Log Files**
  The Log Files icon expands to display icons for all of the log file screens. You can view the System log, the BIG/ip log, or the Pinger log.
- **BIG/config Options**
  Displays the BIG/config options which allow you to customize the BIG/config application window.

Applying changes to the system

When you click the **Apply**, **Add**, or **Delete** buttons, your changes are immediately applied to the system, and they are also saved in the appropriate system configuration file.

Modified configuration files

The BIG/config application modifies the following configuration files:

- */etc/bigip.conf*
- */etc/netstart*
- */etc/bigd.conf*
- */etc/ipfw.conf*
- */etc/ipfwrate.conf*
- */etc/rateclass.conf*
- */etc/bigip.interfaces*
- */etc/hosts.allow*
- */etc/snmpd.conf*

You can modify these files outside of BIG/config, using command line utilities such as BIG/pipe, or using a text editor. However, if you modify configuration files that are not controlled by BIG/ip software utilities, such as those associated with IP filtering and rate filtering, you should be aware that the BIG/config application may not be able to manage configuration files that incorporate complex syntax. Essentially, you run the risk of no longer being able to display or edit the file using the BIG/config application.

Understanding global property settings

In the BIG/config application, you can define four types of global property settings:

- Virtual address properties
- Virtual port properties
- Node address properties
- Node port properties

In the BIG/config application, you can also set specific properties for each virtual server and each node. Some of these properties override the global properties listed above.

You access global property settings from a specific virtual server screen, or from a specific node screen. You access an address' global property settings by clicking the address in the table. Similarly, you access a port's global property settings by clicking the port number shown in the table.

Working with global virtual address properties

Virtual address properties include whether the address is enabled, a connection limit, and the netmask and broadcast address. These properties apply to all virtual servers that use the virtual address, and they correlate to the properties you would otherwise define using the **`bigpipe vip`** command using only the **`<virtual addr>`** parameter (different from commands that use both the **`<virt addr>`** and **`<port>`** parameters, which correlate to the settings you apply to a specific virtual server in BIG/config).

Working with global virtual port properties

Virtual port properties include whether the port is enabled, a connection limit, and timeout settings for inactive connections, connections that use TCP persistence, and connections that use UDP persistence. These properties apply to all virtual servers that use the virtual port, and they correlate to the settings you would otherwise define using the **`bigpipe port`**, the **`bigpipe udp`**, and the **`bigpipe persist`** commands.

Working with global node address properties

Node address properties include whether the node address is enabled, a connection limit, a ratio or priority level for load balancing (applies only if you use Ratio or Priority modes), and a node alias that the BIG/ip Controller can use for node ping to help optimize large configurations (those with 1,000 or more nodes). These properties apply to all nodes that use the node address, and they correlate to the settings you would otherwise define using the **`bigpipe node`** command with the **`<node addr>`** parameter, as well as the **`bigpipe ratio`** and the **`bigpipe alias`** commands.

Working with global node port properties

Node port properties include whether the port is enabled, and they determine how the BIG/ip Controller verifies that the node is up and available to receive connections. If you enable service check, the BIG/ip Controller connects to the node and opens the port to verify that the service on the port is available. If you enable ECV service check, the BIG/ip Controller connects to the node and searches for a user-defined string in the content page returned by the node. These properties apply to all nodes that use the node port number, and they correlate to the settings you would otherwise define using the **`bigpipe node`** command with the **`<node addr>:<port>`** parameter, as well as the **`bigpipe tping_svc`** and the **`bigpipe ssl`** commands. Note that you can override a node port's global extended content verification settings for specific nodes.

Finding help on specific BIG/config screens

Each BIG/config screen provides a **Help** button, which accesses online help for that screen. Online help provides a brief overview of the BIG/config screen, and also provides important information about the syntax required for any configurable settings.

Setting system properties for the BIG/ip Controller

You can view system property settings by clicking BIG/ip in the System tree. The BIG/ip System Properties screen displays information such as the host name and whether the BIG/ip Controller is currently in active or standby mode.

Setting system properties

You can set the following properties in this screen:

- **Load Balancing Method**
  The default load balancing mode is set to Round Robin. You can choose a different load balancing mode from the drop-down list. For information on the supported load balancing modes, refer to Chapter 9.
- **Watch Dog Armed**
  The Watch Dog Armed setting allows you to switch the BIG/ip Controller into and out of fail-safe mode. In fail-safe mode, the BIG/ip Controller acts either as the active unit or the standby unit in a redundant system. On the active unit, the watch dog timer monitors the BIG/ip Controller system and hardware. Should the watch dog timer detect a failure, the unit fails-over to the standby unit.
- **Node ping**
  Node ping sets the BIG/ip Controller to send a standard echo ping to each node address that it manages. If the node address responds to the ping, the nodes associated with the node address are considered *up* and available to accept connections. If the node address does not respond to the ping within the allotted time, the nodes associated with the node address are considered *down* and the BIG/ip Controller does not attempt to send connections to those nodes. To configure node ping, you need to set a frequency and a timeout in the Ping and Timeout boxes.

Setting advanced system properties

In the BIG/ip System Properties screen, you can click the **Advanced Properties** button in the toolbar to display settings for system control variables that affect BIG/ip Controller features. To turn a variable on, check the box. To turn it off, clear the box. You can set the following system control variables:

- Transparent Node Mode
- Rewrite destination address and port on inbound packets
- Allow persistence on virtual servers
- Use persistence as time limit
- Disable IP aliases on virtual addresses
- Forward source routed packets

- IP source checking

Synchronizing configurations in a redundant system

If you are setting up a redundant system, you can configure the virtual servers and nodes on one BIG/ip Controller, and then synchronize the configuration with the other BIG/ip Controller unit. To synchronize the configuration, click the **Synch Configuration** button in the toolbar. Note that this button displays only on the those systems that are configured to allow synchronization (see Chapter 4).

The Synchronize Configuration screen displays the IP address or host name for the unit that you are synchronizing to the current BIG/ip Controller. To perform the synchronization, click **Synchronize Configuration**.

Configuring virtual servers and nodes

There are four basic tasks involved in configuring a virtual server:

- Defining the virtual server's virtual address, port, and the first node to which the virtual server maps.
- Setting properties on the virtual address.
- Setting properties on the virtual port.
- Setting properties on the node, node address, and node port.

Once you configure the virtual server, you can add additional nodes to the virtual server, or remove existing nodes from the virtual server.

Adding a virtual server

Adding a virtual server is a two-part task. First, you define the virtual server itself, and then you may want to set properties on the virtual server, or set global properties on the virtual address or the virtual port that the virtual server uses.

To define the virtual server

1. Click **Virtual Servers** in the System tree.
2. On the Virtual Servers screen, click **Add Virtual Server**.
3. In the Add Virtual Server screen, enter the virtual server's IP address in the **Virtual Address** box.
4. In the **Port** box, either type a port number, or select a service from the drop-down list.
5. In the **Node Address** box, enter the address of the first node to which the virtual server maps.
6. In the **Node Port** box, type the node port number, or select the service from the drop-down list.
7. Click **Add** to send the changes to the BIG/ip system.

Setting properties for the virtual server, the virtual address, and the virtual port

You set specific properties for the virtual server in the Virtual Server Properties screen. From that screen, you can access the global properties for the virtual address and the virtual port used by the virtual server.

To set properties for the virtual server

1. Click **Virtual Servers** in the System tree.
2. On the Virtual Servers screen, click the virtual server for which you want to define properties.
3. In the Virtual Server Properties screen, check **Enabled** to allow the virtual server to accept connections.
4. In the **Connection Limit** box, set a connection limit by entering the maximum number of connections you want to allow on the virtual server at one time. If you do not want to apply a connection limit, set the value to 0.
5. Check **Uses SSL Protocol** to enable SSL persistence.

6. In the **SSL Session ID Persistence** box, set the time allowed for SSL session IDs to be stored on the BIG/ip Controller.
7. In the **SSL Idle Connection Timeout** box, set the time limit for inactive connections to remain connected before being dropped.
8. Click **Apply** to send the changes to the BIG/ip system.

To set global properties on the virtual address

1. In the specific Virtual Server screen, click the virtual address displayed in the **Virtual Address** box.
2. In the Virtual Address Properties screen, check **Enabled** to allow the virtual address to accept connections.
3. In the **Connection Limit** box, set the maximum number of connections you want to allow on the virtual address at one time. If you do not want to apply a connection limit, set the number to 0.
4. In the **Netmask** box, set an alternate netmask only if you do not want to use the default netmask **`255.255.255.0`**.
5. In the **Broadcast** box, set an alternate broadcast only if you do not want to accept the default broadcast. The BIG/ip Controller determines the default broadcast based on the IP address and specified netmask.
6. Click **Apply** to send the changes to the BIG/ip system.

To set global properties on the virtual port

1. In the specific Virtual Server screen, click the virtual port displayed in the **Virtual Port** box.
2. In the Virtual Port Properties screen, check **Enabled** to allow the virtual port to accept connections.
3. In the **Idle Connection Timeout** box, type the number of seconds you want to elapse before an idle connection is dropped.

4. In the **TCP Persist** box, enter the number of seconds for which TCP session information is stored. If you do not want to allow TCP persistence on the port, set the value to 0.
5. In the **UDP Persist** box, enter the number of seconds for which UDP session information is stored. If you do not want to allow UDP on the port, set the value to 0.
6. In the **Connection Limit** box, set the maximum number of connections you want to allow on the virtual port at one time. If you do not want to apply a connection limit, set the value to 0.
7. Click **Apply** to save the changes to the BIG/ip system.

Note

*In the BIG/config application, you do not specifically allow virtual ports. When you create a virtual server that uses a virtual port, the port is both allowed, and enabled, by default. To deny and disable a virtual port in BIG/config, you clear the **Enabled** box.*

Adding nodes to a virtual server

When you define a virtual server, you must define one node to which the virtual server maps. Once you define the virtual server, you can add other nodes to the virtual server mapping.

1. Click **Virtual Servers** in the System tree.
2. On the Virtual Servers screen, select the virtual server for which you want to add a node.
3. In the specific Virtual Server screen, click **Add Node**.
4. On the Add Node screen, in the **Node Address** box, enter the IP address of the node you want to add
5. In the **Node Port** box, enter the node port number which hosts the service provided by the node.
6. Click **Add** to send the changes to the BIG/ip system.

7. After you add a new node, you return to the specific Virtual Server screen. Repeat the process if you want to add more nodes.

Removing nodes from a virtual server

You can remove a node from a virtual server at any time. Note that if you want to remove a node from a virtual server mapping only temporarily, you also have the option of disabling the node by clearing the **Enabled** box on the Node screen, rather than removing the node from the virtual server mapping altogether.

To remove a node from a virtual server mapping

1. Click **Virtual Servers** in the System tree.
2. On the Virtual Servers screen, select the virtual server for which you want to add a node.
3. In the specific Virtual Server screen, the Virtual Server Mapping tables includes a **Remove** button next to each node in the list. To remove the node, simply click **Remove**.

Setting properties for a node, a node address, and a node port

Similar to virtual servers, virtual addresses, and virtual ports, you can set properties for nodes, node addresses, and node ports. Note that you can get to these node properties settings in two different ways:

- You can click **Nodes** in the System tree, and then select a node from the list.
- You can select a specific node from the Virtual Server Mapping table, shown on each individual Virtual Server screen.

The Node screen itself gives you access to the properties for the node address and the node port associated with the node.

To set properties on a node

1. Click **Nodes** in the System tree.

2. On the Nodes screen, select the node for which you want to set properties.
3. In the Node screen, check **Enabled** to allow the node to accept connections.
4. In the **Connection Limit** box, set the maximum number of connections you want to allow on the node at one time. If you do not want a connection limit, set the value to 0.
5. Check **Enabled** in the ECV section to specify that the BIG/ip Controller use Extended Content Verification to determine the node status (whether the node is *up* or *down*).
6. From the **Type** list, select the type of regular expression you want to use for extended content verification:
   - **Normal** looks for content to match the receive rule.
   - **Reverse** looks for content that does not match the receive rule.
   - **SSL** allows the BIG/ip Controller to connect to the port over SSL.
7. In the **Send String** box, enter a regular expression that defines the send information (for information on creating regular expressions for extended content verification, see Chapter 7).
8. In the **Receive Rule** box, enter the receive string. For information on creating regular expressions for extended content verification, see Chapter 7.
9. Click **Apply** to save the changes to the BIG/ip system.

To set global properties on a node address

1. Click **Nodes** in the System tree.
2. On the Nodes screen, select a node which uses the node address.
3. On the specific Node screen, click the node address displayed in the **Address** box.

4. In the Node Address Properties screen, check **Enabled** to allow the node address to accept connections.
5. In the **Connection Limit** box, set the maximum number of connections that you want to allow on the node address at one time. If you do not want to apply a connection limit, set the value to 0.
6. In the **Ratio or Priority** box, enter a number to be used by either the Ratio load balancing mode, or by the Priority load balancing mode, if applicable.
7. In the **Node Alias** box, enter another node address already configured on the BIG/ip Controller to use for node ping. The node address must be an IP alias that points to the same physical server as the node address for which you are setting properties. Note that this setting is used only for optimizing large configurations (those with 1,000 or more nodes). For more information about optimizing large configurations, see Chapter 7.
8. · Click **Apply** to save the changes to the BIG/ip system.

To set global properties on a node port number

When you set global properties on a node port number, all nodes that use the port number inherit the property settings. The global node port property settings apply to all servers that host nodes. Note that you can override the global extended content verification settings for specific nodes.

1. Click **Nodes** in the System tree.
2. On the Nodes screen, select a node which uses the node port number.
3. On the specific Node screen, click the node port number displayed in the **Port** box.
4. In the **Frequency (seconds)** box, enter the interval at which you want the BIG/ip Controller to perform a service check on the node port. If you do not want the BIG/ip Controller to perform a service check on the port, set the value to 0.

5. In the **Timeout (seconds)** box, set the period of time in which the node must respond to the service check in order to be marked *up*.
6. Check **Enabled** in the ECV section to specify that the BIG/ip Controller use extended content verification to determine the node status (whether the node is *up* or *down*).
7. From the **Type** list, select the type of regular expression you want to use for extended content verification:
   - **Normal** looks for content to match the receive rule.
   - **Reverse** looks for content that does not match the receive rule.
8. In the **Send String** box, enter a regular expression that defines the send information. For information on creating regular expressions for extended content verification, see Chapter 7.
9. In the **Receive Rule** box, enter the receive string. For information on creating regular expressions for extended content verification, see Chapter 7.
10. Click **Apply** to save the changes to the BIG/ip system.

Configuring network address translations

You can configure one network address translation for each node address included in the BIG/ip Controller configuration. A network translation address (NAT) provides an alias IP address that a node can use when connecting to clients on the external network. A NAT is also useful if you need remote access to a node via the BIG/ip Controller.

To configure a network address translation

1. Click **NATs** in the System tree.
2. In the Network Address Translations screen, click **Add NAT**.

3. On the Add NAT screen, in the **Node Address** box, enter the node address which you want to associate with the NAT address.
4. In the **NAT Address** box, enter the IP address that you want to use as the node address alias.
5. In the **NAT Netmask** box, change the default netmask only if you do not want to use the default NAT address, 255.255.255.0.
6. In the **NAT Broadcast** box, change the broadcast only if you do not want to use the default broadcast. The BIG/ip Controller automatically determines the NAT broadcast, based on the NAT address and the specified NAT netmask.

Note that you can edit a specific NAT by clicking the NAT address in the Network Address Translations list.

Configuring system redundancy

For each network interface, you can configure special settings for system redundancy, such as turning on the watch dog timer, and configuring the IP alias shared by the two BIG/ip Controllers in the redundant system. You must configure system redundancy settings on both the external and the internal interface.

Using the interface fail-safe option

You can set the BIG/ip Controller to monitor network traffic on the network interface cards (NICs). Should the BIG/ip Controller detect a loss of network traffic on an interface card that exceeds the specified period of time, the BIG/ip Controller initiates a fail-over.

1. Click **NICs** in the System tree.
2. In the Network Interface Cards Properties screen, select the desired interface.

3. In the Redundant System Configuration table, check **Arm Failsafe** to turn on the fail-safe option for the selected interface.
4. In the **Timeout** box, enter the maximum time allowed for a loss of network traffic before a fail-over occurs.
5. In the **Shared IP Alias** box, enter the IP address shared for the corresponding interface on both BIG/ip Controllers.
6. In the **Shared IP Alias Netmask** box, change the shared IP alias netmask only if you do not want to use the default netmask, which is 255.255.255.0.
7. In the **Shared IP Alias Broadcast** box, change the broadcast for the shared IP alias only if you do not want to use the default broadcast.
8. In the **MAC Masquerade** box, enter a shared MAC address only if necessary.
9. Click **Apply** to save the changes to the BIG/ip Controller system.

Be sure to repeat these steps for the second interface. Note that you can synchronize the configuration between two BIG/ip Controllers by clicking the **Synch Configuration** toolbar button in the BIG/ip System Properties screen (see page 4- 4-9).

Configuring IP filters and rate filters

In BIG/config, you can configure simple IP filters and rate classes. If you want to use BIG/config to work with IP filters and rate filters, we recommend that you do not edit the config files associated with these outside of the BIG/config application.

The order in which filters are listed in the IP Filters and Rate Filters tables is important. Filters are applied in a hierarchical order, first to last. You can rearrange the filter order by choosing an action from the Action box, such as **Move down a slot**, or **Move to bottom of the list**.

Configuring IP filters

When you define an IP filter, you can filter traffic in two ways:

- You can filter traffic going to a specific destination or coming from a specific destination, or both.
- The filter can allow network traffic through, or it can deny network traffic.

To define an IP filter

1. Click **IP Filters** on the System tree.
2. In the IP Filters screen, click **Add Filter**.
3. On the Add IP Filter screen, in the **Name** box, type a filter name.
4. From the **Type** list, choose *Accept Packet* to allow traffic, or *Deny Packet* to reject traffic.
5. In the **Source IP Address** box, enter the IP address from which you want to filter traffic, only if you want the filter to be applied to network traffic based on its source.
6. In the **Source Port** box, enter the port number from which you want to filter traffic, only if you want the filter to be applied to network traffic based on its source.
7. In the **Destination IP Address** box, enter the IP address to which you want to filter traffic, only if you want the filter to be applied to network traffic based on its destination.
8. In the **Destination Port** box, enter the port number to which you want to filter traffic, only if you want the filter to be applied to network traffic based on its destination.
9. Click **Add** to add the IP filter to the system.

Configuring rate filters and rate classes

Rate filters are a type of extended IP filter. They use the same IP filter method, but they apply a ***rate class*** which determines the speed of network traffic allowed through the filter. Rate filters are

useful for sites that have preferred clients. For example, an e-commerce site may want to set a higher throughput for preferred customers, and a lower throughput for random site traffic.

Configuring rate filters involves both creating a rate filter and a rate class. When you configure rate filters, you can use existing rate classes. However, if you want a new rate filter to use a new rate class, you must configure the new rate class before you configure the new rate filter.

To configure a new rate class

1. Click **Rate Filters** on the System tree.
2. In the Rate Filters screen, click **Add Class**.
3. On the Rate Class screen, in the **Name** box, type a rate class name.
4. In the **Bits Per Second Allowed** box, enter the maximum number of bits per second that you want the class to allow.
5. In the **Minimum Number of Bits Outstanding** box, enter the minimum number of bits required to be sent for processing from the queue at one time.
6. In the **Queue Length (in Packets)** box, enter the maximum number of packets allowed in the queue. Once the BIG/ip Controller fills the queue, it begins to drop subsequent packets received.
7. Click **Add** to add the rate class to the system.

To configure a rate filter

1. Click **Rate Filters** on the System tree.
2. In the Rate Filters screen, click **Add Class**.
3. On the Rate Filter screen, in the **Name** box, type a rate filter name.
4. From the **Rate Class** list, choose a rate class.

5. In the **Source IP Address** box, enter the IP address from which you want to filter traffic, only if you want the filter to be applied to network traffic based on its source.
6. In the **Source Port** box, enter the port number from which you want to filter traffic, only if you want the filter to be applied to network traffic based on its source.
7. In the **Destination IP Address** box, enter the IP address to which you want to filter traffic, only if you want the filter to be applied to network traffic based on its destination.
8. In the **Destination Port** box, enter the port number to which you want to filter traffic, only if you want the filter to be applied to network traffic based on its destination.
9. Click **Add** to send the changes to the system.

Configuring the BIG/ip SNMP agent

BIG/config allows you to enable the BIG/ip SNMP agent, and it allows you to easily define three aspects of the SNMP agent:

- **Client access**
  You can define an address and netmask for a workstation from which SNMP requests are acceptable.
- **System information**
  You can name a system contact, a machine location, and a community string.
- **Trap configuration**
  You can enter a trap sink, a trap community, and authorize a trap enabled.

You may want to refer to Chapter 6 for more information about the BIG/ip SNMP agent and the MIB.

Configuring SNMP settings

The BIG/config application provides sample SNMP settings for your reference. If you want to use the BIG/ip SNMP MIB, you need to replace these sample settings with settings appropriate to your environment and your specific SNMP package.

To set SNMP properties

1. Click **SNMP** in the System tree.
2. In the BIG/ip SNMP Configuration screen, check **Enabled** to allow access to the BIG/ip SNMP agent.
3. In the **Allow Address** box, enter the address from which the agent can accept requests. Note that you can enter a range of IP addresses, if desired.
4. In the **Allow Netmask** box, enter the netmask from which the agent can accept requests.
5. In the **System Contact** box, enter the SNMP system contact name and email address.
6. In the **Machine Location** box, enter a machine location, such as *First Floor*, or *Building 1*.
7. In the **Community String** box, enter a community string, which is a clear text password used for basic SNMP security.
8. In the **Trap Sink** box, enter the host that should receive trap information.
9. In the **Trap Community** box, enter the community string (password) to use for sending traps.
10. Check **Auth Trap Enabled** to allow traps to be sent for authentication warnings.

Viewing the Extended Content Verification Summary

For convenience, the BIG/config application provides an Extended Content Verification Summary screen, which displays ECV service check settings for all nodes that are set to use ECV service check. To change ECV service check settings for a node, simply click the node entry in the summary table, and then change the settings as desired in the specific Node Properties screen.

Using the BIG/ip System Command for command line access

The BIG/ip System Command screen provides you command line access to the BIG/ip Controller system. You can enter any BIG/pipe command. Usage guidelines and command syntax are provided on the screen. Note that for domestic BIG/ip product packages, the BIG/config application is set to run on port 443, the default SSL port. Any commands you issue in this screen on a domestic system, or on those international systems which customers have equipped with SSL, are sent over a secure connection.

For details on working with BIG/pipe commands, refer to Appendix B.

Viewing system statistics and log files

The BIG/config application allows you to view a variety of system statistics and system log files. Note that from each statistics screen, you can access property settings for individual virtual servers, nodes, IP addresses, and ports by clicking the individual item in the statistics table.

Viewing system statistics

BIG/config allows you to view the following statistical information:

- BIG/ip system statistics, including the elapsed time since the last system reboot, the number of packets and connections handled by the system, and the number of dropped connections.
- Virtual servers, including virtual servers, virtual address only, or virtual ports only.
- Nodes, including nodes, node addresses only, or node ports only.
- NAT statistics, such as the number of packets handled by each NAT.
- IP filter statistics, including the number of packets accepted and rejected by individual IP filters.
- Rate filter statistics, including the number of bits passed through, delayed, and dropped by individual rate filters.
- Information about illegal connection attempts, such as the source IP addresses from which the illegal connection is initiated.

Statistics are displayed in real-time. You can specify the update frequency by setting an interval (in seconds), and then clicking **Update**.

Viewing log files

BIG/config allows you to display three different log files:

- The BIG/ip system log, which displays standard UNIX system events.
- The BIG/ip log, which displays information specific to BIG/ip events, such as defining a virtual server.
- The Pinger log, which displays status information determined by each node ping issued by the BIG/ip Controller.

5

Working With the BIG/pipe Command Line Utility

- **System configuration tasks**
- **Configuring virtual servers and nodes**
- **Configuring BIG/ip system settings**
- **Synchronizing BIG/ip redundant systems**
- **Removing and returning items to service**

System configuration tasks

If you choose to use the BIG/pipe command line utility to do configuration tasks, you need to connect to the BIG/ip Controller via a secure shell, such as the F-Secure SSH client included with the BIG/ip platform, or you need to do the tasks using the VGA monitor and keyboard connected directly to the machine.

If you are making changes to a redundant system, you can make configuration changes to either the active or the standby BIG/ip Controller. We recommend that you make changes to the standby BIG/ip Controller, force a fail-over, and then make the changes on the other machine once it becomes the standby unit.

If you have a single BIG/ip Controller, you may want to put the BIG/ip Controller into maintenance mode before you begin making changes to the system configuration. When the BIG/ip Controller runs in maintenance mode, it does not accept new connections, but it does allow the existing connections to complete. The **`bigpipe maint`** command toggles the BIG/ip Controller in and out of maintenance mode. To put a BIG/ip Controller into maintenance mode, type the following on the command line:

```
bigpipe maint
```

Once you enter the command, the BIG/ip system prompts you to either enter or exit maintenance mode.

Note

If you prefer to configure virtual servers using host names rather than IP addresses, you may first need to define the host names in the /etc/hosts file.

Required tasks for initial configuration

When you first configure a particular BIG/ip Controller, you need to configure the virtual servers and nodes. If you work with a redundant system, you also need to synchronize the configuration between the two machines. The following tasks are required:

- Enable all virtual ports which the virtual servers will use
- Define the virtual servers

- Set properties for virtual servers and nodes
- Set the load balancing mode
- Synchronize redundant systems

Note that the */etc/bigip.conf* configuration file must store the required configuration settings in a specific order (see the following section on working with configuration files). When the BIG/ip Controller reads the configuration file, it has to read the virtual server definitions before it can process connection limit settings for a given virtual server, for example.

If you are changing an existing configuration directly on the command line, instead of in a configuration file, the order in which you make the changes does not matter.

Optional tasks for initial configuration

If you are setting up the initial configuration on a BIG/ip Controller, the following tasks are optional but not required system configuration tasks:

- Configuring Extended Content Verification or Extended Application Verification
- Configuring network address translations for the servers managed by the BIG/ip Controller
- Defining IP filters and rate filters
- Setting up the SNMP agent

You can configure ECV service check and EAV service check using BIG/pipe commands, and you can also define network address translations using BIG/pipe commands. However, to define IP filters and rate filters, or to set up the SNMP agent, you need to use other command line utilities. For more information about these tasks, refer to Chapter 6.

Conventions used in command line syntax

For your convenience, we use typographic conventions to help you identify user input versus computer output or configuration file text, and also to identify parameters that you need to provide when typing commands.

User input and computer output

The Courier typeface is used to distinguish user input and computer output from explanatory text.

User input, such as BIG/pipe commands, is shown in bold Courier type:

```
bigpipe vip
```

Computer prompts and output are shown in plain Courier type:

```
PORT 23     http
(cur, max, tot, reaped) = (0, 0, 0, 0)
```

Mandatory parameters

In command line syntax, angle brackets ("**< >**") enclose mandatory parameters where you must type data associated with a command, such as an IP address or the name of a load balancing mode. For example, when you use the **bigpipe node <ip>** command, you replace the **<ip>** parameter with an IP as shown below:

```
bigpipe node 192.168.1.1
```

Parameter values

When specific parameter values are suggested, such as names of load balancing modes, the acceptable parameter values are separated by a vertical bar (" **|** ").

Note

Appendix B provides a comprehensive BIG/pipe command reference, and you can also find BIG/pipe command reference information in the BIG/pipe man page. To view the BIG/pipe man page, simply type **man bigpipe**.

Working with system configuration files

When you make system configuration changes using BIG/pipe commands, you have two options:

- You can edit a specific configuration file in a text editor such as vi or pico.

- You can enter commands directly at the command prompt and change the currently running system configuration.

When you change the system configuration directly on the command line, your changes are not committed to the system until you specifically save the current configuration. If you reboot or reset the BIG/ip Controller before saving the system configuration, your configuration changes are lost. Note, however, that when you save the current system configuration, you are overwriting the default configuration file, */etc/bigip.conf*.

You may prefer to make changes in a configuration file, which you can easily load into the system, test, edit, and save. Once you validate that configuration file, you can then set it to be the default configuration file.

WARNING

Whether you change the system configuration directly on the command line, or by editing a configuration file, we strongly recommend that you first make a backup copy of your original default system configuration (/etc/bigip.conf), so that you can always return to the original system state.

The default system configuration file

The default BIG/pipe configuration file is */etc/bigip.conf*, and it stores the default BIG/ip configuration which defines all virtual servers. The */etc/bigip.conf* file is created by the First-Time Boot utility, which automatically runs the first time you boot the BIG/ip Controller.

You can change or add virtual servers in the default configuration file, or you can create additional configuration files with virtual servers that you may want to test before implementing in the default configuration file. Editing configuration files is a safe and easy way to implement a configuration change, because you can verify a configuration before committing it to the system. Once you save a new or modified configuration file, you should load it into the system and test it.

Adding and modifying virtual server definitions in a system configuration file

Note that in a BIG/pipe configuration file, you do not need to precede commands with the word "**bigpipe**."

1. Make a backup copy of the existing configuration file so that you can return to your original system configuration at any time.
2. Open an existing configuration file, or open a new file, in a text editor, such as vi or pico.
3. Modify or add **vip** commands in the file using the following syntax, where each **vip** command includes all of the node:port pairs associated with the virtual address:

```
vip <virt addr>:<virt port> define <node addr>:<node port>... \
    <node addr>:<node port>
```

Note: *The* **vip** *command has additional supported syntax, all of which is supported in BIG/pipe configuration files. See Appendix B for details.*

4. Once you have modified or added the desired virtual servers, save the configuration file, and exit out of the text editor.
5. Verify that the new configuration file uses the correct command syntax by typing the following directly on the command line, where **<filename>** is the name of the configuration file:

```
bigpipe -d -f <filename>
```

Loading and testing a system configuration file

The BIG/ip Controller reads configuration files when you boot or reset the system. If you want to test a specific BIG/pipe configuration file, enter the following BIG/pipe command, which resets the system and then automatically loads the named configuration file:

```
bigpipe -f <filename>
```

Note

*When you change the **/etc/bigip.conf** file, you still need to load the configuration file into the system in order for the changes to take effect. If you want to make configuration changes that are effective immediately, enter BIG/pipe commands directly on the command line, and then save the configuration.*

You should test each virtual server that you created or modified. Once you test a configuration file, you can return to the default configuration file simply by resetting the BIG/ip system, which automatically loads the default BIG/pipe configuration file.

If you modified the default configuration file and you want to return to your original configuration, enter the **`bigpipe -f <filename>`** command, using the name of the backup copy of your original configuration file to load your original configuration. Once your original configuration file is loaded into the system, you can enter the following command to save the original configuration file as the default configuration file:

```
bigpipe -s </etc/bigip.conf>
```

Saving a default system configuration file

The BIG/ip Controller always uses the same default configuration file name: */etc/bigip.conf*. To set a specific configuration file to be the default configuration file, you need to load the configuration file using the **`bigpipe -f <filename>`** command, and then save it under the standard default configuration file name using the following command:

```
bigpipe -s </etc/bigip.conf>
```

Modifying the system configuration during runtime

To make changes to the system configuration that are immediately effective, you enter **`bigpipe vip`** commands at the command prompt. It is important to understand that when you modify the system configuration in this way, the modified configuration is used only until the BIG/ip Controller is booted or reset.

If you want to save changes you make to the configuration during runtime, you have two options:

- You can save the current configuration using an alternate configuration file name, such as */etc/test.config*. We recommend saving changes in an alternate configuration file if you do not currently have a backup copy of your default configuration file.
- You can save the current configuration using the default configuration file name (*/etc/bigip.conf*), which overwrites the default configuration file.

WARNING

Because the configuration file is an integral part of the BIG/ip system, we strongly recommend that you make a back-up copy of the original ***/etc/bigip.conf*** *configuration file before you edit or overwrite it.*

To save the current system configuration

The following command saves the current system configuration, including changes you have made during runtime, using the filename you specify:

```
bigpipe -s <filename>
```

Configuring virtual servers and nodes

When you configure virtual servers and nodes, there are certain tasks you must complete before you begin other tasks. For example:

- Before you define virtual servers, you must enable the virtual ports, using the **`bigpipe port`** command, that the virtual servers use.
- A node must be defined as a member of a virtual server before you can work with the node using the **`bigpipe node`** command, or other commands specific to nodes such as **`bigpipe nat`** or **`bigpipe ratio`**.

- If you want to allow UDP connections for any virtual server, you must allow UDP on a specific virtual port.

Viewing the currently defined virtual servers and nodes

When used without any parameters, BIG/pipe commands typically display currently configured elements. For example, the **`bigpipe vip`** command displays all currently defined virtual servers, and the **`bigpipe node`** command displays all nodes currently included in virtual server mappings. The following sections provide BIG/pipe command syntax associated with configuration. For information about using BIG/pipe commands when monitoring your existing system, refer to Chapter 8. For full syntax information on all BIG/pipe commands, see the *BIG/pipe Command Reference* in Appendix B.

Allowing virtual ports and setting virtual port properties

Virtual ports have the following properties that are global, meaning that they apply to all virtual servers that use the virtual port:

- Whether or not the port is allowed
- A maximum number of connections allowed
- A timeout for inactive connections
- TCP persistence
- UDP and UDP persistence

You use the **`bigpipe port`** command to enable or disable virtual ports, and to set connection limits on virtual ports. The **`bigpipe treaper`** command sets a timeout for inactive connections. Once the timeout is exceeded, the BIG/ip Controller drops the inactive connection. The **`bigpipe persist`** command sets TCP persistence, and the **`bigpipe udp`** command controls UDP and UDP persistence.

Allowing a virtual port

By default, all virtual ports on the BIG/ip Controller are denied. If you include a virtual port in any virtual server definition, you must specifically allow the port. To allow virtual ports on the BIG/ip

Controller, use the following command where **<port ID number or name,…>** is a list of the standard port numbers or names for the Internet services you provide:

```
bigpipe port <port> allow
```

Note

In order for FTP to function, you must specifically allow ports 20 and 21 (or ftp and ftp-data). For passive FTP, however, you need only allow port 21.

For example, if you are enabling HTTP (port 80) and telnet (port 23) services, enter the following BIG/pipe command:

```
bigpipe port 80 23 allow
```

or

```
bigpipe port www telnet allow
```

For FTP, you must always use ports 20 and 21 (ftp and ftp-data) together. For example, to configure the BIG/ip Controller to serve HTTP and FTP, use:

```
bigpipe port 80 20 21 allow
```

If you want to deny a previously allowed virtual port, use the following command syntax:

```
bigpipe port <port> deny
```

Setting a connection limit on a virtual port

The **bigpipe port** command also sets a connection limit for a port using the following parameters:

```
bigpipe port <port>... <port> limit <limit>
```

For example, the following command limits the number of connections allowed virtual port 80 to 5,000:

```
bigpipe port 80 limit 5000
```

Setting timeouts for inactive connections

The **bigpipe treaper** command sets a timeout, in seconds, for which an inactive connection is allowed to continue. Once the timeout expires, the inactive connection is immediately dropped. You can define a inactive connection timeout for one or more ports using the following command:

```
bigpipe treaper <port>... <port> <seconds>
```

For example, the following command sets a 1200 second time limit for inactive connections on port 443:

```
bigpipe treaper 443 1200
```

Setting persistence for TCP connections

If a virtual server that uses the virtual port requires persistence for TCP connections, you need to specifically enable TCP persistence for that virtual port. Note that all virtual servers which use the virtual port inherently allow TCP persistence. Essentially, the **bigpipe persist** command enables persistence at the same time as setting the persistence time limit. You can set TCP persistence for one or more virtual ports at a time:

```
bigpipe persist <port>... <port> <seconds>
```

The following sample command sets TCP persistence on ports 80 and 443, and allows persistent connection information to be stored for one hour:

```
bigpipe persist 80 443 3600
```

Allowing UDP connection and setting UDP persistence values

You must specifically allow UDP connections on virtual ports that need to support UDP traffic. The **bigpipe udp** command allows you to allow for UDP connections, and it also sets the time that UDP connection information is stored. You can set UDP persistence on one or more ports at a time:

```
bigpipe udp <port>... <port> <seconds>
```

The following command allows UDP connections on port 5050, and it stores UDP connection information for 5 minutes:

```
bigpipe udp 5050 300
```

To disable UDP on a port, set UPD persistence to zero:

```
bigpipe udp 5050 0
```

Defining virtual servers and setting virtual server properties

The basic command that defines virtual servers is **bigpipe vip**. The **bigpipe vip** command supports several parameters that allow you to define a number of aspects of virtual servers including the nodes to which the virtual server maps, the number of connections allowed on the virtual server, and whether or not the virtual server allows SSL persistence.

If a site provides multiple services, you need to define a separate virtual server for each service. For example, if a web site is meant to support both HTTP and SMTP email, you need to define two different virtual servers as follows:

```
bigpipe vip www.SiteOne.com:http define node1:http node2:http
bigpipe vip mail.SiteOne.com:smtp define node1:smtp node2:smtp
```

The virtual port used by the virtual server does not necessarily have to match the port numbers used by each of the nodes. For example, the following command routes web traffic destined for the HTTP port to the nonstandard port, 8001, on each of the 2 nodes:

```
bigpipe vip www.SiteOne.com:http define node1:8001 node2:8001
```

Note

*When you work with BIG/pipe commands, you can substitute domain names for virtual IP addresses and host names for physical IP addresses (as long as the host and domain names are defined in your **/etc/hosts** file), and you can also substitute port numbers with standard service names, such as http or ftp.*

There are certain parameters associated with the **bigpipe vip** command that set properties for individual elements of the virtual server, such as the virtual address. Some properties that are associated with an element of the virtual server may affect other virtual servers that use the same element. For example, if you set a custom netmask for a virtual address, all virtual servers that use that

address also use that netmask. Note that you cannot set properties for a virtual address until the virtual address is defined in one or more virtual servers.

Defining the virtual server mapping

A virtual server often maps to more than one node, and you will likely have multiple nodes associated with any given virtual server. You can configure a complete virtual server, including multiple node mappings, using a single command:

```
bigpipe vip <virtual addr>:<virtual port> define \
   <node addr>:<node port>... <node addr>:<node port>
```

For example, the following command defines a virtual server that offers mail service on port 25:

```
bigpipe vip mail.SiteOne.com:smtp define node1:smtp define \
   node2:smtp node3:smtp
```

The virtual server mapping shown above maps the *mail.SiteOne.com* virtual server to three different nodes.

If you want to remove a node from a virtual server mapping, you essentially have to redefine the virtual server mapping without the node you want to remove. You may find it easier to use the BIG/config application to remove nodes from existing virtual servers because you do not have to redefine the virtual server. You simply select the node you want to remove.

Setting a connection limit on the virtual server

The **bigpipe vip** command also sets a connection limit for the virtual server. As with most BIG/pipe commands, you can include one or more virtual servers in the single command. The command uses the following parameters, where **<limit>** is the maximum number of connections allowed on the virtual server at one time:

```
bigpipe vip <virtual address:virtual port>... \
   <virtual address:virtual port> limit <limit>
```

If you want to remove a connection limit from an existing virtual server, set the **<limit>** parameter to 0.

Setting SSL persistence on a virtual server

The **bigpipe vip** command also configures SSL persistence for a specific virtual server. The command uses the following parameters:

```
bigpipe vip <virtual addr>:<port> define <node addr>:<node<port> \
   special <protocol> <persistence timeout> \
   <inactive connection timeout>
```

The **<protocol>** parameter should be set to **ssl**. The **<persistence timeout>** parameter is set in seconds, and the **<inactive connection timeout>** is also set in seconds. Note that the **<inactive connection timeout>** parameter applies only to inactive SSL connections, and this setting override the setting defined by the **bigpipe treaper** command that controls inactive connection timeout for all connection types.

Note

You can define SSL settings only when you define the virtual server. If you want to activate SSL on an existing virtual server, or remove it from an existing virtual server, you must redefine the virtual server using the new SSL settings, or leaving SSL out of the virtual server definition altogether.

Setting properties for a virtual address

Once you define a specific virtual address in a virtual server, you can set global properties for that virtual address that apply to all virtual servers which use it. The global property settings simply include:

- Whether or not the virtual address is enabled
- A maximum number of connections allowed on the virtual address
- A custom netmask and broadcast address

You set virtual address properties using the **bigpipe vip** command where the first parameter is the virtual address alone, rather than the virtual address followed by the virtual port. Using the parameters shown below, you can enable or disable one or more virtual addresses at a time:

```
bigpipe vip <virtual addr> ... <virtual addr> enable
bigpipe vip <virtual addr> ... <virtual addr> disable
```

To define a connection limit for a virtual address, use the **<limit>** parameter as shown below:

```
bigpipe vip <virtual addr> ... <virtual addr> limit <limit>
```

The default netmask for any virtual address is set to `255.255.255.0`. You can apply a custom netmask to a virtual address when you define a virtual server that uses the virtual address. Note that the netmask applies to all virtual servers which use the virtual address.

```
vip <virtual addr>:<virtual port> netmask <netmask> define \
   <node addr>:<node port>... <node addr>:<node port>
```

The default broadcast address is based on the virtual IP address and the currently specified netmask. You can set a different broadcast address, if required, by issuing the following command (note that you must include the netmask definition in the command):

```
bigpipe vip <virtual addr>:<virtual port> broadcast <broadcast> \
   define <node addr>:<node port>... <node addr>:<node port>
```

If you want to set both a custom netmask and a custom broadcast for a virtual address, you must define both when you define a virtual server that uses the virtual address:

```
bigpipe vip <virtual addr>:<virtual port> netmask <netmask> \
   broadcast <broadcast> define <node addr>:<node port>... \
   <node addr>:<node port>
```

Setting properties for a node

Nodes support several properties, including a connection limit, and Extended Content Verification. For individual port numbers used in node configuration, you can set global properties, such as the interval at which the BIG/ip Controller performs a service check.

These settings are used by all nodes that incorporate that port number. You can also set properties for node addresses which apply to all nodes that use those node addresses.

Note

You can not set properties for nodes, node addresses, or node ports which are not currently included in at least one virtual server mapping.

Verifying services on a node port

There are three ways in which you can have the BIG/ip Controller verify services on nodes which use a specific port number:

- **Service check**
  Service check simply requires that the BIG/ip Controller connect to the port and establish a connection with the service that the node supports.
- **ECV service check**
  ECV service check uses the Extended Content Verification feature. When using ECV service check, the BIG/ip Controller looks for a user-specified string in the content that the service first returns.
- **EAV service check**
  EAV service check uses the Extended Application Verification feature. Essentially, EAV service check performs the same function as ECV service check, except that it allows a custom external checker program to determine whether or not a specific service or specific site content is available on the node.

These settings are global, and they apply to all nodes that use the node port number. On specific nodes, however, you can override the global settings for Extended Content Verification and customize the send and receive strings.

The **`bigpipe tping_svc`** command sets the interval, in seconds, at which the BIG/ip Controller verifies whether a service on a node is available or not. This command applies to all nodes which use the specified node port. Also note that the interval which you

define with this command is used for all types of service verification including service check, ECV service check, and EAV service check. The syntax for the command is:

bigpipe tping_svc <port> <seconds>

The **bigpipe timeout_svc** command sets the time allowed, in seconds, for nodes to respond to a service check, ECV service check, or EAV service check. If a specific node does not respond within the time limit, the BIG/ip Controller automatically marks the node *down*. This command also applies to all nodes which use the specified node port number:

bigpipe timeout_svc <port> <seconds>

Defining send strings and receive rules for Extended Content Verification

The send strings and receive rules that you define for Extended Content Verification are actually defined in the */etc/bigd.conf* file, and not defined by BIG/pipe commands. When a */etc/bigd.conf* file is present, the BIG/ip Controller searches the file for a node port, or a specific node. If the BIG/ip Controller does not find the node port, or the node itself, in the file, it performs a basic service check. If it does find the node or node port defined, it performs the service check using Extended Content Verification. Note that the BIG/ip system does not include a default */etc/bigd.conf* file; you have to create one if you choose to use ECV service check.

Note

The BIG/ip Controller reads the /etc/bigd.conf file only at startup, or when the bigd daemon is restarted. If you edit the /etc/bigd.conf file, you need to reboot the BIG/ip Controller, or restart the bigd daemon by typing **bigd** *on the command line.*

In the */etc/bigd.conf* file, you can define a global send string and receive rule for each node port number. All nodes which use that port number use the global send string and receive rule, unless you to define a specific send string and receive rule for a specific node.

To define a send string and receive rule for a global node port number, you need to include the following line in the file:

active <port> <send_string> <receive_string>

To define a send string and receive rule for a specific node, you need to include the following line in the file:

active <node addr>:<node port> <send_string> <receive_string>

If you don't specify a **<send_string>**, the BIG/ip Controller uses a default send string, **Get /**, which returns the home page when sent to a web server. The **<receive_string>** is a POSIX regular expression (see the man page for details). If you do not specify a receive string, the BIG/ip Controller considers any string received to be a match. This can create inaccurate results, because an HTML page that returns a "404 Not Found" error would actually be considered to be a match if the user did not specify a particular receive string. There is a 5000 byte limit on receive strings.

For more information about using Extended Content Verification, see Chapter 7.

Note

The /etc/bigd.conf file can contain only one send string and receive rule for each port, or for each specific node.

Using inverted regular expressions for Extended Content Verification

When you define send and receive strings in the */etc/bigd.conf* file, you can use a special syntax to allow for inverted regular expressions. If you set up Extended Content Verification using inverted regular expressions, the BIG/ip Controller marks the node *down* if it matches the send string you specify. If the content does not match the send string, then the BIG/ip Controller marks the node *up*.

In each string definition, you need to replace the **active** keyword with the **reverse** keyword:

reverse <port> <send_string> <receive_string>

reverse <node addr>:<node port> <send_string> <receive_string>

For example, the following lines incorporate an inverted regular expression:

active node1:80 "GET /" "html"

```
reverse node2:80 "GET /" "error"
```

If the content that the BIG/ip Controller retrieves contains the word "error," the BIG/ip Controller considers it a match to the specified receive string, and it marks the node *down*.

Using Extended Content Verification on an SSL connection

When you define send and receive strings in the */etc/bigd.conf* file, you can use a special syntax to allow the send and receive strings to be sent over an SSL connection. In each string definition, you need to replace the **`active`** keyword with the **`ssl`** keyword:

```
ssl <port> <send_string> <receive_string>
ssl <node addr>:<node port> <send_string> <receive_string>
```

The BIG/ip Controller uses SSL version 3, as do popular web browsers, but it automatically falls back to SSL version 2 when it connects to web servers that support only version 2.

Enabling a node and setting a node connection limit

The **`bigpipe node`** command allows you to enable a node, and also allows you to define a connection limit for the node. To enable or disable one or more nodes, use the command with the following parameters:

```
bigpipe node <node addr>:<node port>... \
   <node addr>:<node port> enable
bigpipe node <node addr>:<node port>... \
   <node addr>:<node port> disable
```

When you disable an existing node, the BIG/ip Controller does not allow new connections to be sent to the node, but it does allow the node to finish processing current connections before it completely takes the node down for service.

You can also set a connection limit using the **`bigpipe node`** command with the following parameter, where **`<limit>`** is the number of connections allowed on the node at one time:

```
bigpipe node <node addr>:<node port> limit <limit>
```

Setting properties for a node address

Node addresses have five global settings which apply to all nodes that use the node address:

- Whether or not the node is enabled
- A connection limit
- A ratio proportion or a priority level
- An IP alias to ping (for use with large configurations)

The **bigpipe node** command allows you to enable a node address, and it also allows you to set a connection limit for the node address.

```
bigpipe node <node addr>... <node addr> enable
bigpipe node <node addr>... <node addr> disable
```

When you disable an existing node address, the BIG/ip Controller does not allow new connections to be sent to any node that uses that node address, but it does allow the nodes to finish processing current connections before it removes them from service.

You also set the connection limit using the **bigpipe node** command with the following parameters, where **<limit>** is the number of connections allowed on the node address at one time:

```
bigpipe node <node addr> limit <limit>
```

The **bigpipe ratio** command defines the value associated with both the Ratio and the Priority load balancing modes. If you select the Ratio load balancing mode, the value is used as the ratio proportion, and if you select the Priority load balancing mode, the value is used as the priority level.

```
bigpipe ratio <node addr>... <node addr> <value>
```

The **bigpipe alias** command allows you to use a node alias for node ping. This option is used only in large configurations that have 1,000 nodes or more, because it reduces network traffic and it does not waste processing resources on the BIG/ip Controller. It also prevents individual servers from being repeatedly pinged on several IP alias addresses where a ping on only one IP alias address would sufficiently determine whether or not all IP aliases on the server are available.

```
bigpipe alias <node addr>... <node addr> pingnode <ip alias>
```

Note that the **<ip alias>** parameter must be set to a node address that is defined in at least one of the virtual servers managed by the BIG/ip Controller.

To remove a node alias, use the following command:

```
bigpipe alias <node addr>... <node addr> delete
```

For more information about this and other issues involved with large configurations, refer to Chapter 7.

Defining network address translations for nodes

You can define one network address translation (NAT) for each node address included in a virtual server mapping. A NAT provides an external IP address used to access or identify the node to outside clients. When a BIG/ip Controller receives a connection request for a specific NAT, it sends the connection directly to the node associated with the NAT, rather than load balancing the connection request across the array of nodes.

A NAT must use a unique IP address that is not used by any other virtual or physical server in your network. The **bigpipe nat** command defines a NAT for a specific node address. All nodes that use the node address also use the associated NAT.

```
bigppipe nat <node addr> to <network translation addr>
```

The NAT definition can also include a custom netmask and broadcast address. The default netmask is set to 255.255.255.0.

```
bigppipe nat <node addr> to <network translation addr> \
   netmask <netmask> broadcast <broadcast>
```

To delete a network address translation for a node, use the following command:

```
bigppipe nat <node addr> delete
```

Note

Nodes that have NATs configured can make requests to virtual servers managed by the BIG/ip Controller, as well as to other NAT addresses managed by the BIG/ip Controller. If a specific node makes a request to a virtual server managed by the BIG/ip Controller, the request is treated as a normal connection request and load balanced across the nodes as normal.

Configuring BIG/ip system settings

The BIG/pipe command line utility provides commands for using the following features on the BIG/ip system:

- The load balancing mode
- Node ping
- Maintenance mode

These features apply to the BIG/ip system as a whole, and they affect all virtual servers and nodes configured on the BIG/ip system.

Setting a load balancing mode

The **`bigpipe lb`** command sets the load balancing mode, and the **`<mode>`** parameter specifies the name of the load balancing mode you want to use:

`bigpipe lb <mode>`

Table 5.1 displays the acceptable values for the **`<mode>`** parameter.

Command	Description
`bigpipe lb rr`	Sets load balancing to Round Robin mode
`bigpipe lb ratio`	Sets load balancing to Ratio mode
`bigpipe lb priority`	Sets load balancing to Priority mode
`bigpipe lb least_conn`	Sets load balancing to Least Connection mode
`bigpipe lb fastest`	Sets load balancing to Fastest mode
`bigpipe lb observed`	Sets load balancing to Observed mode
`bigpipe lb predictive`	Sets load balancing to Predictive mode

Table 5.1 *Command syntax for setting load balancing mode*

For detailed information about how each of the BIG/ip Controller's load balancing modes distributes connections, refer to Chapter 9.

WARNING

If you set the load balancing mode to Ratio or Priority, you must define the ratio or priority settings for each node address. The value you define using the **`bigpipe ratio`** *command is used as the ratio value if Ratio is the currently selected load balancing mode, and the same value is used as the priority level if Priority is the currently selected load balancing mode. See the* ***Setting properties for a node address*** *section on page 5 - 5-20.*

Configuring node ping

The BIG/pipe utility provides two commands for controlling node ping:

```
bigpipe tping_node <seconds>
bigpipe timeout_node <seconds>
```

The **`bigpipe tping_node`** command sets the interval at which the BIG/ip Controller performs a ping on each node address it manages. The **`bigpipe timeout_node`** sets the number of seconds that server associated with the node address has to respond to the ping. If the server responds to the ping within the timeout period, the BIG/ip Controller marks the nodes associated with that node address as *up*. If the server does not respond within the timeout period, the BIG/ip Controller marks the nodes associated with the node address as *down*.

The default value for **`tping_node`** is 5 seconds, and the default value for **`timeout_node`** is 15 seconds. Using the default settings, the BIG/ip Controller pings each node every 5 seconds and if it does not receive a response in 15 seconds, it marks the pinged node as being *down*.

Setting both **tping_node** and **timeout_node** to 0 seconds disables node ping.

WARNING

If you disable node pinging, you run the risk of permanently losing nodes marked as down at the time you disable node pinging. If a node is marked as being down, the BIG/ip Controller marks it as being up only when the BIG/ip Controller can successfully ping the node and receive a response.

Node ping modes

There are three modes of node ping: ICMP, TCP, and none.

- **ICMP**
  This mode is helpful in troubleshooting network problems. The BIG/ip Controller sends an ICMP echo packet to each node address in intervals determined by the **tping_node** setting. If the BIG/ip Controller receives an ICMP echo reply packet from the server associated with the node address before the timeout elapses, the BIG/ip Controller marks the node as *up*.
- **TCP**
  This mode is useful if one, or more, of the servers is not capable of replying to ICMP pings. The BIG/ip Controller attempts to connect to TCP port 7 (the standard "echo" service) on each node in intervals determined by the **tping_node** setting. If the BIG/ip Controller connects successfully, it writes a few bytes to the port, and then immediately tries to read the same bytes back on the same connection. If the BIG/ip Controller successfully reads and writes the bytes before the timeout elapses, it marks the node as *up*. Note that for TCP node ping to work, all nodes must be configured to respond to it.
- **None**
  In this mode, no node ping is performed. All nodes are considered *up*.

Note

The mode you choose applies to all nodes. For example, you cannot choose TCP for one node and ICMP for another.

Selecting a node ping mode

The BIG/ip Controller uses the ICMP node ping mode as the default node ping mode. You can change the node ping mode to use TCP ping, or you can turn node pinging off, by adding a **`bigdflags`** setting in the */etc/netstart* file on the BIG/ip Controller:

- TCP
  `bigdflags=-s`
- None
  `bigdflags=-n`

To return the node ping mode to the default setting (ICMP), simply remove the **`bigdflags`** setting from the */etc/netstart* file.

Synchronizing BIG/ip redundant systems

The **`bigpipe configsync`** command simplifies the process of propagating configuration changes to the second BIG/ip Controller in a BIG/ip redundant system. Use this command after you change the kernel configuration on one of the units in the redundant system. The **`bigpipe configsync`** command writes the current configuration to the */etc/bigip.conf* file. If SSH RSA Authentication is properly configured between the two BIG/ip units, the command then copies */etc/bigip.conf* on the local BIG/ip Controller to the */etc/bigip.conf* file on the remote BIG/ip Controller, and then loads the new configuration file to the kernel on the remote BIG/ip Controller.

WARNING

The **`bigpipe configsync`** *command overwrites the default configuration file on both the BIG/ip Controllers with the current configuration of the BIG/ip Controller from which you execute the command. We recommend that you make backup copies of the configuration files on both systems before executing this command.*

The command syntax is simply:

bigpipe configsync

Note that the **bigpipe configsync** command is a shortcut for the following commands:

```
bigpipe -s /etc/bigip.conf
scp /etc/bigip.conf root@<ip-addr>:/etc/bigip.conf
ssh -l root <ip-addr> /sbin/bigpipe -f /etc/bigip.conf
```

Using the interface fail-safe option

For maximum reliability, the BIG/ip platform supports fail-over detection on both its internal and external interface cards. When you arm the fail-safe option on the interface cards, the BIG/ip Controller monitors network traffic going through the interfaces. Should it detect a loss of traffic on either interface, the BIG/ip Controller fails over to the standby unit.

- **External interface**
  The fail-safe option on the external interface listens for traffic going to the BIG/ip Controller. If the BIG/ip Controller does not detect traffic for a period time equal to half the fail-safe timeout, the BIG/ip Controller attempts to generate network traffic by issuing ARP requests for the default router. If the BIG/ip Controller does not detect traffic before the fail-safe timeout elapses, the BIG/ip Controller fails over to the standby unit.
- **Internal interface**
  The fail-safe option on the internal interface also listens for traffic going to the BIG/ip Controller. If the BIG/ip Controller does not detect traffic for a period time equal to half the fail-safe timeout, the BIG/ip Controller attempts to generate network traffic by issuing ICMP echo requests to each node included in its configuration. Any traffic on the interface, including replies to the ICMP echo ping, averts a fail-over. However, if the BIG/ip Controller does not detect traffic before the fail-safe timeout elapses, the BIG/ip Controller fails over to the standby unit. Note that if the BIG/ip Controller configuration does not include any nodes, the BIG/ip Controller operates as if the internal interface is not armed for fail-safe.

Arming fail-safe on an interface

The **bigpipe interface** command displays the current fail-safe settings, and also allows you to change arm or disarm the fail-safe on a particular interface.

Each interface card installed on the BIG/ip Controller has a unique name, which you need to know when you set the fail-safe option on a particular interface card. To view the names of both interface cards installed in the BIG/ip Controller, type the following command:

```
bigpipe interface
```

To arm fail-safe on a particular interface, type the following command:

```
bigpipe interface <ifname> failsafe arm
```

To disarm fail-safe on a particular interface, type the following command:

```
bigpipe interface <ifname> failsafe disarm
```

For example, say you have an external interface named exp0 and an internal interface named exp1. To arm the fail-safe option on both cards, you need to issue the following two commands:

```
bigpipe interface exp0 failsafe arm
bigpipe interface exp1 failsafe arm
```

WARNING

You should arm fail-safe on an interface only once the BIG/ip Controller is in a stable production environment. Otherwise, routine network changes may cause fail-over unnecessarily.

Setting a specific BIG/ip Controller to be the preferred active unit

In a redundant configuration, you can set a specific BIG/ip Controller to be the preferred active unit. Any time the preferred unit is operational, it runs as the active BIG/ip Controller. We recommend that you use this feature only in special situations. For example, if you are testing upgraded BIG/ip software on one unit,

but running a previous BIG/ip software version on the other unit, you may want to set the BIG/ip Controller running the new software as the preferred active unit.

To set a specific unit as the active unit, you need to modify **sod** settings in the */etc/rc.local* file on both BIG/ip Controllers in the redundant system.

1. Open the */etc/rc.local* file on the preferred active unit.
2. Find the following entry in the file:

```
echo " sod.";  /usr/sbin/sod 2> /dev/null
```

3. Change the entry to read:

```
echo " sod.";  /usr/sbin/sod -force_master 2> /dev/null
```

4. Save and close the file.
5. Open the */etc/rc.local* file on the preferred standby unit.
6. Find the following entry in the file:

```
echo " sod.";  /usr/sbin/sod 2> /dev/null
```

7. Change the entry to read:

```
echo " sod.";  /usr/sbin/sod -force_slave 2> /dev/null
```

8. Reboot both BIG/ip Controllers.

Note

This change applies only when you reboot the BIG/ip Controller. Therefore, if you issue the **bigpipe fo slave** *command on a preferred active machine, the active machine reverts to standby state as expected.*

For more information about using fail-over commands in BIG/pipe, refer to Appendix B.

Removing and returning items to service

Once you have completed the initial configuration on the BIG/ip Controller, you may want to temporarily remove specific items from service for maintenance purposes. For example, if a specific network server needs to be upgraded, you may want to disable the nodes associated with that server, and then enable them once you finish installing the new hardware and bring the server back online.

If you specifically disable the nodes associated with the server, the BIG/ip Controller allows the node to go down only after all the current connections are complete. During this time, the BIG/ip Controller does not attempt to send new connections to the node. Although the BIG/ip Controller's monitoring features would eventually determine that the nodes associated with the server are down, specifically removing the nodes from service prevents interruptions on client connections.

You can remove the entire BIG/ip Controller from service, or you can remove the following individual items from service:

- Virtual servers
- Virtual addresses
- Virtual ports
- Nodes
- Node addresses

Removing the BIG/ip Controller from service

The BIG/ip platform offers a Maintenance mode, which allows you to remove the BIG/ip Controller from network service. This is useful if you want to perform hardware maintenance, or make extensive configuration changes. When you activate Maintenance mode, the BIG/ip Controller no longer accepts connections to the virtual servers it manages. However, the existing connections are allowed to finish processing so that current clients are not interrupted.

The **`bigpipe maint`** command toggles the BIG/ip Controller into or out of Maintenance mode. The command syntax is simply:

`bigpipe maint`

If the BIG/ip Controller runs in Maintenance mode for less than 20 minutes and you return the machine to the normal service, the BIG/ip Controller quickly begins accepting connections. However, if the BIG/ip Controller runs in Maintenance mode for more than 20 minutes, returning the Controller to service involves updating all network ARP caches. This process can take a few seconds, but you can speed the process up by reloading the */etc/bigip.conf* file using the following command:

```
bigpipe -f /etc/bigip.conf
```

Removing individual virtual servers, virtual addresses, and ports from service

The BIG/ip Controller also supports taking only select virtual servers, addresses, or ports out of service, rather than removing the BIG/ip Controller itself from service. Each BIG/pipe command that defines virtual servers and their components supports **`enable`** and **`disable`** keywords, which allow you to remove or return the elements from service:

When you remove a virtual address or a virtual port from service, it affects all virtual servers associated with the virtual address or virtual port. Similarly, if you remove a node address from service, it affects all nodes associated with the node address.

Enabling and disabling virtual servers and virtual addresses

The **`bigpipe vip`** command allows you to enable or disable individual virtual servers, as well as virtual addresses. To enable or disable a virtual server, type the appropriate command:

```
bigpipe vip <virtual addr>:<virtual port> enable
bigpipe vip <virtual addr>:<virtual port> disable
```

To enable or disable a virtual address, type the appropriate command:

```
bigpipe vip <virtual addr> enable
bigpipe vip <virtual addr> disable
```

Enabling and disabling virtual ports

The **bigpipe port** command allows you to allow or deny traffic on a virtual port:

```
bigpipe port <virtual port> allow
bigpipe port <virtual port> deny
```

Removing individual nodes and node addresses from service

Enabling and disabling nodes and node addresses

The **bigpipe node** command allows you to enable or disable individual nodes, as well as node addresses. To enable or disable a node, type the appropriate command:

```
bigpipe node <node addr>:<node port> enable
bigpipe node <node addr>:<node port> disable
```

To enable or disable a node address, type the appropriate command:

```
bigpipe node <node addr> enable
bigpipe node <node addr> disable
```

6

Additional System and Network Configuration

- **Changing passwords for the BIG/ip Controller**
- **Editing the /etc/hosts file**
- **Configuring Sendmail**
- **Configuring the BIG/ip SNMP agent**
- **Enabling dynamic routing**
- **Configuring the BIG/ip Controller for DNS proxy**
- **Configuring DNS resolution**
- **Converting from rotary DNS**

Changing passwords for the BIG/ip Controller

During the First-Time Boot utility, you define a password that allows remote access to the BIG/ip Controller, and you also define a password for the BIG/ip web server. You can change these passwords at any time.

Changing the BIG/ip Controller password

1. At the BIG/ip Controller command line prompt, login as root and use the **passwd** command.
2. At the password prompt, enter the password you want to use for the BIG/ip Controller and press Return.
3. To confirm the password, retype it and press Return.

Changing passwords and adding new user IDs for the BIG/ip web server

You can create new users for the BIG/ip web server, change a password for an existing user, or recreate the password file altogether, without actually going through the BIG/ip web server configuration process.

Creating new users and changing passwords for existing users

The following command creates a new user ID, or changes the password for an existing user ID. In place of the **<username>** parameter, enter the user ID for which you want to create a password:

```
/var/f5/httpd/bin/htpasswd /var/f5/httpd/basicauth/users \
   <username>
```

Once you enter the command, you are prompted to enter the new password for the named user.

Creating a new password file

The following command recreates the BIG/ip web server password file, and defines one new user ID and password. In place of the **<username>** parameter, enter the user ID that you want to create:

```
/var/f5/httpd/bin/htpasswd -c /var/f5/httpd/basicauth/users \
   <username>
```

Once you enter the command, you are prompted to enter the new password for the named user.

Editing the /etc/hosts file

The First-Time Boot utility configures the initial */etc/hosts* file. You can made additions and edits to this file as desired. To make configuration of virtual servers and nodes easier, we suggest that you define host names for each virtual address and node address that you plan on using in your configuration.

The default */etc/hosts* file that the First-Time Boot utility creates looks similar to the file displayed in Sample Screen 6.1 shown below. Note that the internal and external network interface addresses shown in the file correspond to those addresses that you entered during the First-Time Boot utility.

```
#bigip host table (default)
127.0.0.1 localhost localhost.host.domain
#add your default gateway here
207.17.112.254
# real - external interface
207.17.112.230 bigip ext
# real - internal interface
192.168.1.100 int
#VIPs (add as necessary)
#nodes (add as necessary
```

Sample Screen 6.1 *The /etc/hosts file as created by the First-Time Boot utility*

Configuring Sendmail

You can configure the BIG/ip Controller to allow electronic mail to be sent from the system. This configuration must be completed if the BIG/ip Controller is to send electronic mail to the administration workstation or to an alphanumeric pager. The BIG/ip platform includes an example configuration file that should be suitable for most sites. Before you use this configuration file, however, you do have to customize it for your network environment.

Customizing the /etc/sendmail file

When you customize this file, you enter the name of the mail relay server.

Finding the mail relay in your network

1. From a machine capable of name resolution, type the following on the command line:

   bigip: /etc# nslookup

2. The command returns a default server name and corresponding IP address:
   `Default Server: <server name>`
   `Address:  <server`

3. Next, query for the mail relay server for your domain using the following commands:

   set q=mx
   <domain name>

The information returned includes the name of the mail exchanger.

Setting up Sendmail

1. Copy */etc/sendmail.cf.off* to */etc/sendmail.cf*.

2. Edit */etc/sendmail.cf* and set the DS variable to the name of the mail exchange server.

3. Open the */etc/crontab* file, and change the last line of the file to read:

```
0,15,30,45 * * * * root /usr/sbin/sendmail -q > /dev/null 2>&1
```

   Including this line in the */etc/crontab* file sets Sendmail to flush the outgoing message queue for any email that could not be delivered immediately. Because the BIG/ip Controller does not accept email from external sources, there is no need to run the Sendmail daemon. Queue flushes are issued via *crontab.*

4. Save and close the */etc/crontab* file.
5. Open the */etc/aliases* file.
6. In the */etc/aliases* file, create an entry for **`root`** to point to an administrator at your site. For example:

   `root: networkadmin@SiteOne.com`

   Because the BIG/ip Controller does not accept local email, bounces or undelivered messages go unnoticed. This requires that the administrator is notified when a message is bounced or undelivered.

7. Save and close the */etc/aliases* file.
8. Run the **`newaliases`** command to generate the new aliases database using the information you just added.
9. Reboot the BIG/ip Controller.

Configuring the BIG/ip SNMP agent

The BIG/ip platform includes a private BIG/ip SNMP MIB. You can configure the SNMP settings in the BIG/config application, or on the command line.

Downloading the MIB

SNMP management software requires you to use the MIB files associated with the device. You may obtain two MIB files from the BIG/ip directory */usr/contrib/f5/mibs*. The *F5LABS-MIB.txt* file defines all the properties associated with F5 specific functionality (load balancing, NATs, etc.), while the *UCD-SNMP-MIB.txt* file specifies general information about the system. For information about the objects defined in the MIB, refer to the descriptions in the OID section of the MIB file.

Understanding configuration file requirements

You need to make changes to several configuration files on the BIG/ip Controller before you use the SNMP agent. Once you change these configuration files, you need to restart the SNMP agent.

/etc/hosts.deny

This file must be present to deny by default all TCP connections to the agent. The contents of the files are as follows:

```
ALL : ALL
```

/etc/hosts.allow

This file is used to configure TCP wrappers. TCP wrappers do basic checking on the source IP and try to verify that the request is legitimate. The basic syntax is as follows, where `daemon` is the name of the daemon, and `IP/MASK` specifies the network that is allowed access:

```
daemon: IP/MASK
```

For example, you might use the following line which sets the Bigsnmpd daemon to allow connections from the **`128.95.46.0/255.255.255.0`** address:

```
bigsnmpd: 128.95.46.0/255.255.255.0
```

The example above allows the 256 possible hosts that are at the network address **`128.95.46.0`** to access the SNMP daemon. Additionally, you may use the keyword `ALL` in any of the fields to allow access for all hosts or all daemons.

/etc/snmpd.conf

The *snmpd.conf* file controls most of the SNMP daemon. This file is used to setup and configure certain traps, passwords, and general SNMP variable names. A few of the necessary BIG/ip variables are listed below:

- **System Contact Name**
  The System Contact is a MIB-II simple string variable defined by almost all SNMP boxes. It usually contains a user name, as well as an email address. This is set by the key `syscontact`.
- **Machine Location (string)**
  The Machine Location is a MIB-II variable that almost all boxes support. It is a simple string that defines the location of the box. This is set by the key `syslocation`.
- **Community String**
  The community string clear text password is used for basic SNMP security. This also maps to VACM groups, but for initial read/only access, it is limited to only one group.
- **Trap Configuration**
  Trap configuration in version 1.0 is done by controlling three properties in *snmpd.conf*:
  - `trapsink HOST`
    This sets the host to receive trap information. `HOST` is an IP address.
  - `trapcommunity STRING`
    This sets the community string (password) to use for sending traps. If set, it also sends a trap upon startup `coldStart(0)`.
  - `authtrapenable INTEGER`
    Setting this variable to 1 enables traps to be sent for authentication warnings. Setting it to 2 disables it.

/etc/netstart

To automatically start the SNMP agent, you must uncomment the line in the **/etc/netstart** file which starts **bigsnmpd**.

/etc/snmptrap.conf

This configuration file includes OID, trap, and regular expression mappings. The configuration file specifies whether to send a specific trap or not based on a regular expression. An excerpt of the config file is shown in Sample Screen 6.2.

```
OID REGULAR EXPRESSION DESCRIPTION
.1.3.6.1.4.3375.1.1.110.6 (ROOT LOGIN) ROOT LOGIN
.1.3.6.1.4.3375.1.1.110.5 (denial) REQUEST DENIAL
.1.3.6.1.4.3375.1.1.110.1 (your expression) Your expression
```

Sample Screen 6.2 *Excerpt from the /etc/snmptrap.conf file*

Some of the OIDs have been permanently mapped to BIG/ip specific events. You may, however, insert your own regular expressions and map them to the 110.1 OID. This is a generic OID for miscellaneous events. When lines match your expression, they will be sent to your management software with the 110.1 OID.

Syslog

You must configure Syslog to send syslog lines to `checktrap.pl` if the syslog lines might make a match and thus form a valid SNMP trap. The following line in the */etc/syslog.conf* file requires that syslog look at every piece of information logged, scan the *snmptrap.conf* file, and determine if a trap should be generated:

```
*.* | exec /sbin/checktrap.pl
```

This trapping mechanism uses fewer syslog resources if it is set up so as not to use `*.*`. More specific priorities and facilities generate less execs to `checktrap.pl`.

Enabling dynamic routing

The BIG/ip platform includes the *GateD* daemon, which is disabled by default. To enable the BIG/ip Controller to accept dynamic routing updates from your routers, you must first create the appropriate configuration file, */etc/gated.conf*. For complete details on configuring the *GateD* daemon, refer to the *GateD User's Guide*, available on the BIG/ip administrative web server (connect to the administrative web server and click the *Online Documentation* link on the first page).

Enabling the GateD daemon

You enable the *GateD* daemon on the BIG/ip Controller by typing the following at the command line prompt:

```
bigip# gated
```

Edit the /etc/netstart file

Next, you need to edit the */etc/netstart* file and change the definition of the gated variable as shown below:

```
gated=YES
```

The BIG/ip Controller is now configured to accept dynamic route updates from your router.

Note

Certain network environments may require that you modify the routing tables or your router. If you have communication problems between your router and the BIG/ip Controller, please contact Technical Support at F5 Labs, Inc.

Configuring the BIG/ip Controller for DNS proxy

You can configure the BIG/ip Controller as a DNS proxy or forwarder. This is useful for providing DNS resolution for servers and other equipment behind the BIG/ip Controller that might want to lookup a domain name or IP address.

To configure DNS proxy, you simply create a */etc/named.boot* file that contains only two lines:

```
forwarders <DNS_SERVERS>
options forward-only
```

In place of the **<DNS_SERVER>** parameter, use the IP addresses of one or more properly configured name servers that have access to the Internet.

You can also configure BIG/ip Controller as an authoritative nameserver for one or more domains. This is useful when DNS is needed in conjunction with phony domain names and network numbers for the servers and other equipment behind the BIG/ip Controller. Please refer to BIND documentation at *http://www.isc.org/bog-4.9.4/bog.html* for complete details.

Configuring DNS resolution

To use fully qualified domain names rather than IP addresses on the BIG/ip Controller, you must create an */etc/resolv.conf* file. The file should have the following format:

```
nameserver <DNS_SERVER_1>
search <DOMAIN_NAME_1> <DOMAIN_NAME_2>
```

In place of the **<DNS_SERVER_1>** parameter, use the IP address of a properly configured name server that has access to the Internet. You can specify two additional name servers as backups, by inserting an additional **nameserver** line for each backup name server.

If you configure the BIG/ip Controller itself as a DNS server, then we suggest that you choose its loopback address (**`127.0.0.1`**) as the first nameserver in the */etc/resolv.conf* file.

Converting from rotary DNS

If your network is currently configured to use rotary DNS, your node configuration may not need modification. However, you need to modify your DNS zone tables to map to a single IP address instead of to multiple IP addresses.

For example, if you had two Web sites with domain names of *www.SiteOne.com* and *www.SiteTwo.com*, and used rotary DNS to cycle between two servers for each Web site, your zone table would look like this:

```
www.SiteOne.com  IN A   192.168.1.1
                 IN A   192.168.1.2
www.SiteTwo.com  IN A   192.168.1.3
                 IN A   192.168.1.4
```

With a the BIG/ip Controller configuration, the IP address of each individual node used in the original zone table becomes hidden from the Internet. It is recommended that you use the Internet reserved address range as specified by RFC 1918 for your nodes: it is no longer necessary to use "real" IP addresses assigned to you by your Internet Service Provider (ISP). In place of multiple addresses, simply use a single virtual server associated with your site's domain name.

Using the above example, under a the BIG/ip Controller configuration your DNS zone table would look like this:

```
www.SiteOne.com  IN A   207.17.112.231
www.SiteTwo.com  IN A   207.17.112.232
```

The IP addresses used above for *www.SiteOne.com* and *www.SiteTwo.com* are virtual addresses associated with specific virtual servers managed by the BIG/ip Controller.

7

Advanced Configurations

- **Working with advanced configurations**
- **Optimizing large configurations**
- **Balancing and managing connections for routers and router-like devices**
- **Using Extended Content Verification**
- **Using an Extended Application Verification program**

Working with advanced configurations

The BIG/ip Controller supports a variety of advanced configuration options and features. There are three types of advanced configurations that you can work with:

- Large configurations that include 5,000 or more virtual servers and nodes.
- ***Extended Application Verification***, which requires that the BIG/ip Controller connects to a specific node and performs a user-defined function to verify that certain applications or data are available on the node.
- ***Transparent Node Mode***, which allows a BIG/ip Controller to manage and load balance connections for network devices, such as transparent firewall or cache servers.

Advanced configurations require special planning, and they introduce specific installation and configuration issues that you do not normally address in a standard BIG/ip Controller configuration.

Optimizing large configurations

The BIG/ip Controller supports up to 40,000 virtual servers and nodes combined. Larger configurations on a BIG/ip Controller, such as those that exceed 1,000 virtual servers or 1,000 nodes, introduce special configuration issues. To ensure a high performance level, you need to change certain aspects of the BIG/ip Controller's management of virtual servers and nodes.

Reducing ARP traffic on the external network

In normal configurations, the BIG/ip Controller maintains an IP alias on its external interface for each virtual address that is managed. IP aliases are broadcast on the network when a virtual server is defined, and also each time a BIG/ip Controller switches from standby mode to active mode in a redundant configuration. In BIG/ip Controller configurations that have thousands of virtual servers defined, the IP aliasing of those servers may lead to a

significant increase in network traffic. Each time a new IP alias is defined, the router on the external network must issue an ARP request for that virtual server's address. This type of configuration also increases fail-over recovery time in BIG/ip redundant systems. When a fail-over occurs, the BIG/ip Controller that becomes the active machine creates an IP alias for each virtual server that it manages. Normally, this process takes less than one second. However, if the BIG/ip Controller has upwards of 8,000 virtual servers, this process can take as long as 90 seconds. The active BIG/ip Controller is unresponsive during the time it creates the IP aliases, and it cannot begin processing connections until the IP aliasing is complete.

To ensure a fast fail-over process, and to help reduce the amount of ARP requests a router must make, you should run the BIG/ip Controller in VIP-NoArp mode. In VIP-NoArp mode, the BIG/ip Controller does not create IP aliases for virtual servers. Instead, network traffic bound for virtual servers configured on the BIG/ip Controller are routed using the BIG/ip Controller's external interface as a gateway. Configuring VIP-NoArp mode is a two-step process:

- On the router, you must configure a gateway to the virtual servers using the BIG/ip Controller's external interface IP address.
- On the BIG/ip Controller itself, you must change the **vip_no_arp** system control variable. Note that you can use either the BIG/config application, or the BIG/pipe command line utility, to change system control variables.

Note

You can enable VIP-NoArp mode only if you have the ability to add a route to your router. Note that in redundant systems, you need to use the shared external IP address as the gateway address for the virtual servers configured on the BIG/ip Controller.

Configuring the router

In the router configuration, you need to define a static route as the gateway for each virtual address managed by the BIG/ip Controller. The static route should set the gateway address to the IP address for the external interface on the BIG/ip Controller. For example, if the

shared external address of a BIG/ip redundant system is 11.0.0.100, and all virtual servers configured on the BIG/ip redundant system use IP addresses 11.0.1. 50 through 11.0.1.55, you need to configure the router to use 11.0.0.100 as a gateway to the 11.0.1.* subnet. Such a definition on a UNIX router would read:

```
route add -net 11.0.1.0 gw 11.0.0.100
```

Activating VIP-NoArp mode in BIG/config

In the BIG/config application, the VIP-NoArp mode setting is under BIG/ip **`sysctl`** configuration. To turn the VIP-NoArp mode on, simply check the **Disable IP Aliases on Virtual Servers** box. To turn VIP-NoArp mode off, clear the **Disable IP Aliases on Virtual Servers** box.

WARNING

We recommend that you do not toggle this mode on or off while the virtual servers are defined. Resetting the variable at that time may result in system anomalies.

Activating VIP-NoArp mode on the command line

You can activate VIP-NoArp mode in one of two ways:

- You can edit the */etc/rc.sysctl* file in an editor, and then reboot the system, which implements the change.
- You can immediately enable or disable the mode using sysctl commands.

If you choose to edit the */etc/rc.sysctl* file, you simply need to add the following line to the file to activate VIP-NoArp mode:

```
sysctl -w bigip.vipnoarp=1
```

To deactivate VIP-NoArp mode, you can either comment the line out, or delete it from the */etc/rc.sysctl* file altogether. Once you edit the file, the changes do not take affect until you reboot the system.

To immediately activate VIP-NoArp mode, type the following on the command line:

```
bigpipe -f /dev/null
sysctl -w bigip.vipnoarp=1
```

```
bigpipe -f /etc/bigip.conf
```

To immediately deactivate VIP-NoArp mode, type the following on the command line:

```
bigpipe -f /dev/null
sysctl -w bigip.vipnoarp=0
bigpipe -f /etc/bigip.conf
```

WARNING

We recommend that you do not toggle the VIP-NoArp mode on or off while the virtual servers are defined. Resetting the sysctl variable at that time may lead to a system crash.

Reducing the number of node pings and service checks issued by the BIG/ip Controller

The BIG/ip Controller checks node status at user-defined intervals in two different ways:

- The BIG/ip Controller can issue a ***node ping*** to all node addresses that it manages. If the BIG/ip Controller receives a response to a node ping from a specific node address, all nodes associated with that node address are marked *up* and available for connections. The node ping can be either ICMP or TCP.
- The BIG/ip Controller can also perform a ***service check***. For each node that uses service check, the BIG/ip Controller connects to the node and attempts to establish a connection with the service configured on the node port. If the BIG/ip Controller is able to establish a connection with the service, the BIG/ip Controller marks the node *up*. If the BIG/ip Controller cannot establish a connection with the service, the BIG/ip Controller marks the node *down*. It is important to note that the node is marked *down*, even if the node's address is able to respond to the BIG/ip Controller's simple node ping.

If a BIG/ip Controller's configuration includes thousands of nodes, the node pings and service checks begin to take up more resources on both the BIG/ip Controller and the servers than is preferred. You can significantly reduce the number of node pings and service checks in configurations that have a group of node addresses which

are all IP aliases on the same server. For each group of node addresses that points to a given server, you can select one node address out of the group to represent all node addresses in the group. The representative node address is referred to as the ***node alias***. When the BIG/ip Controller issues a node ping or service check, it sends the ping or performs the service check only on the node alias, rather than on all nodes in the group. If the BIG/ip Controller receives a valid response before the timeout expires, it marks all nodes associated with the node alias as *up* and available to receive connections. If the BIG/ip Controller does not receive a valid response before the timeout expires, it marks all of the nodes associated with the node alias as *down*.

An important note about service checks

You can set the BIG/ip Controller to use a node alias for nodes that are configured for service check; however, there are some limitations to this implementation. Service checks are port-specific, unlike node pings which are merely sent to a node address. If you assign a node alias to a node that uses service check, the node alias must be configured to support the port number associated with the node. If the node alias is not configured properly, the BIG/ip Controller can not establish a conversation with the service that the specific node supports, and the service check is invalid.

Note

If you have configured different ports on each node to handle a specific Internet service and you want to use IP aliases, you can use BIG/pipe commands to work around the situation. Refer to the BIG/pipe Command Reference in Appendix B for more information about the **`bigpipe alias`** *command.*

Setting up node aliases in BIG/config

In the BIG/config application, each node address has a set of properties associated with it, including the **Node Alias** property. Note that before you define a node alias for a specific node address, you may want to check the properties for each node that uses the

node alias. The node alias must support each port used by a node that is configured for service check, otherwise the service check results are invalid.

1. Select **Nodes** in the System tree to display the Virtual Servers page.
2. In the Node Properties table, click the node address.
3. In the Node Address Properties page, type the node alias in the **Node Alias** box.
4. Click **Apply**.

Setting up node aliases using the BIG/pipe command line utility

The BIG/pipe command line utility allows you to set node aliases for multiple nodes at one time. With the **`bigpipe alias`** command, you can do three things:

- View all node aliases defined in the current configuration
- View the node alias associated with a specific node address
- Define a node alias for one or more node addresses

For details about working with the **`bigpipe alias`** command, refer to the *BIG/pipe Command Reference* in Appendix B.

Balancing and managing connections for routers and router-like devices

To provide for load balancing across transparent network devices, you have to run the BIG/ip Controller in a special mode, called ***Transparent Node Mode***. In Transparent Node Mode, the BIG/ip Controller appears, to its clients, to be a router, which handles all traffic going to the external network. In Transparent Node Mode, a node address is actually the next-hop address to which the BIG/ip Controller routes packets, and a node port is the port that the BIG/ip Controller checks to determine whether or not a specific service on the node is *up* or *down*.

If you run Transparent Node Mode, you need to configure special types of virtual servers on the BIG/ip Controller called ***wildcard virtual servers***. A ***default wildcard virtual server*** accepts connections where the IP address and port number does not match any other IP address and port number defined as a virtual server. ***Port-specific wildcard virtual servers*** accept connections where the IP address does not match any virtual server, but the connection requests a port associated with a particular port-specific wildcard virtual server.

Installation and configuration issues

Running a BIG/ip Controller in Transparent Node Mode introduces several configuration issues, including the following:

- You need to connect the BIG/ip Controller's external interface to the network where the clients reside, which in this case is an internal network. The BIG/ip Controller's internal interface needs to be connected to the network where the array of routers or firewalls sits, which is typically thought of as external to the rest of the network.
- You need to set the default route on the transparent network devices appropriately.
- You need to activate transparent node mode, a BIG/ip system control property.
- You need to define at least one wildcard virtual server, either a default wildcard virtual server, or a port-specific wildcard virtual server.
- You need to specifically enable each virtual port used by a wildcard virtual server.

Connecting the BIG/ip Controller to the network

If you choose to use Transparent Node Mode on the BIG/ip Controller, the BIG/ip Controller internal and external interfaces are not connected to the network the same way in which they would be in a standard configuration. Instead, the BIG/ip Controller's external interface, which receives connection requests from clients, is connected to the network where the client workstations reside. In

Transparent Node Mode, clients are internal workstations that request connections to sites outside of the network; thus the BIG/ip Controller's external interface is actually connected to the internal network.

The BIG/ip Controller's internal interface is connected to the same logical network as the internal interface of the routers or transparent firewalls that the BIG/ip Controller load balances. The transparent device's external interface should be connected to the external network in the same fashion as it would be in a standard network configuration.

Figure 7.1 shows a BIG/ip redundant system that is connected and configured to handle transparent node mode for two transparent firewalls.

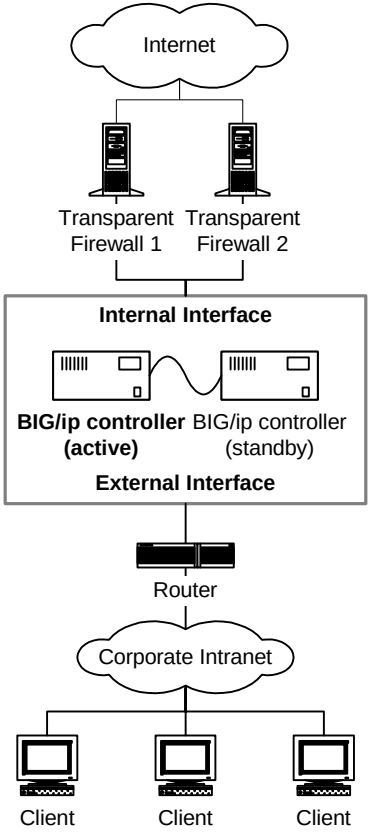

Figure 7.1 *Transparent node mode configuration*

Also note that when you set up a BIG/ip Controller for Transparent Node Mode, you need to set the default route on internal routers to the BIG/ip Controller's external interface IP address.

Configuring the BIG/ip Controller in Transparent Node Mode

Configuring the BIG/ip Controller for Transparent Node Mode is similar to setting up a standard BIG/ip configuration, and it involves three basic tasks:

- Activating transparent node mode
- Defining one or more wildcard virtual servers
- Allowing traffic on each virtual port associated with a wildcard virtual server

You can configure these settings either in the BIG/config application, or in the BIG/pipe command line utility. In Transparent Node Mode, you can use any BIG/config options or individual BIG/pipe commands that you normally use to control virtual servers, virtual ports, and nodes.

Activating Transparent Node Mode

Transparent Node Mode is a BIG/ip system control property. You can easily activate Transparent Node Mode using the BIG/config application. The BIG/ip system control properties are available from the System Tree in the BIG/config application. To toggle a particular system control property, you simply check or clear the corresponding box. You can also set the system control property using the Sysctl utility.

Setting the system control variables in BIG/config

In BIG/config, system control variables are displayed in the BIG/ip Advanced Properties page. To get to the Advanced Properties page, click the **BIG/ip** in the System Tree, and then click **Advanced Properties** at the top of the BIG/ip System Properties page.

To turn Transparent Node Mode on, you actually need to set two different system control properties in the BIG/config application:

- Check **Transparent Node Mode** to activate Transparent Node Mode.

- If you previously enabled the IP source checking system control variable, disable it by clearing the **IP sourcecheck** box.

Note

The default setting for IP source checking is disabled.

Setting the system control variables using Sysctl

You can also use the Sysctl command line utility to change the system control properties. To view the currently selected mode from the command line, type:

```
sysctl bigip.bonfire_mode
```

To activate Transparent Node Mode, type:

```
sysctl -w bigip.bonfire_mode=1
```

To return the BIG/ip Controller to normal mode, type:

```
sysctl -w bigip.bonfire_mode=0
```

If you previously enabled the IP source check system control variable, you need to disable it:

```
sysctl -w net.inet.ip.sourcecheck=0
```

To permanently save the currently selected mode, you need to save the */etc/rc.sysctl* file. You can also make these changes by editing the file manually in a text editor. For more information about working with the Sysctl utility on the BIG/ip platform, refer to Appendix C.

Creating a wildcard virtual server

Normally, a BIG/ip Controller directs traffic based on matching the requested IP address to a virtual address defined for one or more virtual servers. In Transparent Node Mode, however, the BIG/ip Controller receives connection requests that have destination IP addresses which are not managed by the BIG/ip Controller. In order to provide for this, Transparent Node Mode supports wildcard virtual servers which accept all traffic that has a requested an IP address that is not defined for any other virtual server in the BIG/ip Controller's configuration. The BIG/ip Controller passes this type of connection request to one of the transparent devices in the array.

In Transparent Node Mode, you can work with two types of virtual servers:

- A ***port-specific wildcard virtual server*** accepts all traffic that does not match the IP address of any other virtual server defined on the BIG/ip Controller, but does include a specific port number which is managed by the port-specific wildcard virtual server. Using port-specific wildcard virtual servers allows you to configure the BIG/ip Controller to balance traffic for a certain port to a specific group of devices.
- A ***default wildcard virtual server*** accepts all traffic that does not match the IP address of any virtual server defined on the BIG/ip Controller, nor does it match a virtual port number used in any port-specific wildcard virtual servers.

All wildcard virtual servers must use a specific wildcard address of **`0.0.0.0`**. A default wildcard virtual server uses port 0, thus its full IP and port address is **`0.0.0.0:0`**. Port-specific wildcard virtual servers must use the **`0.0.0.0`** wildcard address, but they can use any virtual port number. If you create port-specific wildcard virtual servers, the BIG/ip Controller uses its standard service check feature to determine whether the specific port on each transparent device is *up* or *down*.

When the BIG/ip Controller receives connection requests in Transparent Node Mode, it first attempts to match the destination IP address to an IP address associated with one or more virtual servers. If there is no match, then the BIG/ip Controller attempts to match the incoming request against a port-specific wildcard virtual server, if you have any defined. Finally, if the BIG/ip Controller does not find a match for a specified port number, it sends the connection to the default wildcard virtual server, or it denies the connection request if there is no default wildcard virtual server defined.

Note

*You cannot define a wildcard IP address, nor can you enable virtual port **0**, unless the BIG/ip Controller is currently running in Transparent Node Mode.*

Defining nodes for a wildcard virtual server

When you define nodes for wildcard virtual servers, you need to use the internal addresses of the transparent devices. Note that the BIG/ip Controller does not translate port numbers when running in Transparent Node mode. Instead, the BIG/ip Controller uses the port number associated with each node to determine the port on which it should perform a service check to determine the node's status (whether the device is *up* or *down*).

The BIG/ip Controller's default node ping setting is ICMP ping. Some transparent devices may not be configured to accept ICMP pings. If the devices in your environment cannot be configured to respond to ICMP pings on their internal ports, you have two options:

- You can switch to TCP Echo ping.
- You can disable node ping entirely.

If you disable node ping entirely, you may want to set the global properties for each node port to use service check. Service check confirms that the BIG/ip Controller can connect to a node port and establish communication with the service managed on that port. If there is no appropriate port on the device, you should disable service check as well.

Configuring routes for Transparent Node Mode

You can configure the BIG/ip Controller to run a routing daemon, *GateD*, or to simply use default and static routes. Aside from the normal interface routes that the operating system automatically creates, the BIG/ip Controller needs only gateway routes to the internal networks (networks inside the firewall), to which the BIG/ip Controller is not directly connected. The BIG/ip Controller must use its external interface to reach these gateways. Note that the BIG/ip Controller does not need any routes to the nodes specified in the default wildcard virtual server.

Using conventional virtual servers in Transparent Node Mode

You can configure conventional virtual servers to handle traffic that needs to be routed to non-transparent devices. This feature is useful in resolving the following issues:

- Some client web browsers may be configured to use a non-transparent proxy.
- Certain email peers may be configured to use an SMTP gateway that is on the firewall. In this case, you may want to add only one firewall node to the virtual server in order to avoid maintaining two or more email configurations.
- You may want to load balance client connections that go to internal network servers.

Using FTP in Transparent Node Mode

A default wildcard virtual server (**`0.0.0.0:0`**) does not handle FTP connection requests. If you need to accommodate FTP connection requests, you should configure two FTP-specific wildcard virtual servers: **`0.0.0.0:20`** and **`0.0.0.0:21`**. Note that the BIG/ip Controller supports connections for non-default active ports on FTP proxy servers.

Printing the connection table

The BIG/pipe command line utility also offers a useful diagnostic tool that prints the list of current connections. Normally, the **`bigpipe dt`** command prints the client, virtual server, and node addresses. In Transparent Node Mode, the **`bigpipe dt`** command also prints the final destination address.

Using Extended Content Verification

Extended Content Verification (ECV) is a sophisticated type of service check typically used to confirm whether or not a node returns specific data upon request. If a node returns the requested data in response to the service check, the BIG/ip Controller marks the node *up*. If the node does not return the requested data, the BIG/ip Controller marks the node *down*.

ECV service checks are based on regular expressions, including a send string and a receive rule. Typically, an ECV service check looks for specific text in an HTML page. For example, you can use Extended Content Verification to search for the name of your company which is listed on the home page for your web site. If the BIG/ip Controller finds the company name on the page returned by the node, it marks the node *up*. If the HTML page instead returns a 404 error, the BIG/ip Controller does not detect a match, and it marks the node *down*.

Formatting the /etc/bigd.conf file

The BIG/ip Controller performs Extended Content Verification for those nodes listed in the */etc/bigd.conf* file. The BIG/ip platform does not include this file; you must create the file yourself. You can either create the file in a text editor, or you can fill in the ECV settings in the BIG/config application for nodes, and for global node port properties.

When you configure ECV service checks, you essentially edit the */etc/bigd.conf* file to specify which nodes to verify, which strings to send, and which regular expressions to match against the received data. The file format is as follows:

```
active <port | service> ["<send_string>" ["<recv_string>"]]
```

Comments start with "**#**" and run to end of line. Blank lines are ignored. You may use single quotes, double quotes, or curly braces to enclose the send string and receive string values. If you do not specify a send string, the BIG/ip Controller uses the default send string:

```
"GET /"
```

When the BIG/ip Controller sends the **"GET /"** to a web server, the server returns the front page for site that it hosts.

If you don't specify a receive rule, however, the BIG/ip Controller considers any data received to be a match. In this case, the BIG/ip Controller marks the node *up* based on whether or not the node returns an HTML page. However, this is not a good service check, because the BIG/ip Controller may inadvertently receive an HTML page that contains error information, such as a "404 Not Found" error, rather than actual site content.

Using the following sample send and receive strings, the BIG/ip Controller performs the following functions:

- Attempts to retrieve a web page called /test.html from each web server and search for the string "site ok" in that page.
- Attempts to connect to the mail daemon on each node and, without sending anything, expects to read a string containing the word "Sendmail".
- Attempts to connect to a web server on a non-standard port, 8000, retrieve a web page called "/", without regard to its contents.
- Attempt to connect to the "finger" port, 79, and query about user "webmaster".

```
active http "GET /test.html" "site ok"
active smtp "" "Sendmail"
active 8000
active 79 "webmaster"
```

The */etc/bigd.conf* file is read once at startup. If you change the file, you must reboot or restart **bigd** for the changes to take effect. To restart **bigd**, use the command **/sbin/bigd [options]**. The new bigd automatically replaces any previous version of **bigd** that was executing.

The BIG/ip Controller continues to read data until the service check closes the connection, or until the data read reaches 5000 bytes, whichever comes first. When picking a search string, pick one that appears in the first 5,000 bytes of the web page.

To test configuration file syntax, you can run **/sbin/bigd -d** in an interactive shell. This command parses the file, compiles any regular expressions, reports any errors, and then exits.

Writing simple regular expressions

Regular expressions are used with ECV service check to determine if a service is functioning properly. For example, if the BIG/ip Controller performs a service check on a web server, the server might return one page if things are properly working, and another page if it is not. Regular expression syntax is fundamentally complex and confusing. Fortunately, it is also possible to write simple regular expressions.

Note

Regular expression syntax is not the same as the "wildcard syntax" that is commonly used in command shells. Also, case treatment is ignored when matching regular expressions.

Examples

To match any received data that starts with "<HEAD>", use:

```
"^<HEAD>"
```

To match any data that contains the string "Welcome To SiteOne", use:

```
"Welcome To SiteOne"
```

To match any data (which has the same effect as not specifying a string), use:

```
""
```

To match any properly formed HTML header, use:

```
"<HEAD>.*</HEAD>"
```

For complete details on regular expressions, refer to the standard, POSIX 1003.2 Section 2.8, or the manual page, available online on the BIG/ip Controller by running:

```
man re_format
```

Using an Extended Application Verification program

Extended Application Verification (EAV) is a sophisticated type of service check typically used to confirm whether an application running on a node is responsive to client requests. To determine whether a node application is responsive, the BIG/ip Controller uses a custom program referred to as an ***external service checker***. An external service checker program essentially provides completely customizable service check functionality for the BIG/ip Controller. It is external to the BIG/ip system itself, and is usually developed by the customer. For example, you can use an external service checker to verify Internet or intranet applications, such as a web application that retrieves data from a back-end database and displays the data in an HTML page.

An external service checker program works in conjunction with the Bigdnode daemon, which verifies node status using node pings and service checks. If you configure external service check on a specific node, the Bigdnode daemon checks the node by executing the external service checker program. Once the external service checker executes, the Bigdnode daemon looks for output written by the external service checker. If the Bigdnode daemon finds output from the external service checker, it marks the node *up*. If it does not find output from the external service checker, it marks the node *down*. Note that Bigdnode does not actually interpret output from the external service checker; it simply verifies that the external service checker created output.

Note

External service checker programs are custom programs that are developed either by the customer, or by the customer in conjunction with F5 Labs.

Configuring EAV service checks

There are four steps to implementing EAV service checks on the BIG/ip Controller:

- Verify that your external service checker program meets certain requirements, such as creating a *pid* file.
- Install the external service checker program on the BIG/ip Controller.
- Allow EAV service checks in the BIG/ip configuration.
- Configure the specific nodes to use EAV service check.

External service checker requirements

Extended Application Verification is intended to provide maximum flexibility. The external service checker programs that you create can use any number of methods to determine whether or not a service or an application on a node is responsive. The external service checker must, however, meet the following minimum requirements:

- The external service checker must use a *pid* file to hold its process ID, and the pid file must use the following naming scheme:
  `/var/run/pinger.<ip>..<port>.pid.`
- As soon as the external service checker starts, if the *pid* file already exists, the external service checker should read the file and send a SIGKILL to the indicated process.
- The external service checker must write its process ID to the *pid* file.
- If the external service checker verifies that the service is available, it must write standard output. If the external service checker verifies that the service is unavailable, it cannot write standard output.
- The external service checker must delete its *pid* file before it exits.

The BIG/ip platform includes a sample external service checker for your reference in the following location:
`/usr/local/lib/pingers/sample_pinger`

The sample external service checker provides a very simple program, shown in Sample Screen 7.1.

```
# these arguments supplied automatically for all external
   pingers:
# $1 = IP (nnn.nnn.nnn.nnn notation or hostname)
# $2 = port (decimal, host byte order)
# $3 and higher = additional arguments
#
# In this sample script, $3 is the regular expression
#

pidfile="/var/run/pinger.$1..$2.pid"

if [ -f $pidfile ]
then
   kill -9 `cat $pidfile` > /dev/null 2>&1
fi

echo "$$" > $pidfile

echo "GET /" | /usr/local/lib/pingers/nc $1 $2 2> /dev/null | \
   grep -E -i $3 > /dev/null

status=$?
if [ $status -eq 0 ]
then
    echo "up"
fi
rm -f $pidfile
```

Sample Screen 7.1 *A sample external service checker program*

Installing the external service checker on the BIG/ip Controller

The */usr/local/lib/pingers* directory is the default location for external service checker applications. You can install external service checker applications to other directory locations if desired.

Allowing EAV service checks

Once you install an external service checker on the BIG/ip Controller, you need to add an entry to the */etc/bigd.conf* file. The standard syntax of the */etc/bigd.conf* file includes the following lines:

```
active  [<node_ip>:]<port> ["<send_string>" ["<recv_pattern>"]]
reverse [<node_ip>:]<port> ["<send_string>" ["<recv_pattern>"]]
ssl     [<node_ip>:]<port> ["<send_string>" ["<recv_pattern>"]]
```

To allow external service checking, you need to add the following entry to the */etc/bigd.conf* file:

```
external  [<node_ip>:]<port> [ <path> ]["<argument_string>"]
```

The **<path>** variable can be an absolute or a relative path to the external checker application. Absolute paths should begin with a slash ("/"). Other paths are relative to the standard pinger directory, */usr/local/lib/pingers*.

The **"<argument_string>"** variable must consist of exactly one quoted string. The string may include any number of arguments, delimited in the usual way by white space, for example:

```
active n1:80 "GET /" "html"
external n1:8000 "my_pinger -a 600 -b"
```

In the above example, the BIG/ip Controller uses plain HTTP to check port 80, but executes */usr/local/lib/pingers/my_pinger* to check port 8000 and supplies it three arguments in addition to the standard arguments.

For another example, say there are three nodes on which the BIG/ip Controller checks port 8000. The BIG/ip Controller executes a separate copy of the external service checker named *my_pinger* for each node:

```
external n1:8000 "my_pinger -a -b"
```

```
external 8000 "my_pinger -b"
```

In this example, the first entry specifies how to ping port 8000 on node n1. The second entry specifies how to ping port 8000 on any other node.

Executing the external service checker program

The BIG/ip Controller performs the external service check at set intervals. The BIG/ip Controller actually uses the service ping interval, which you set using the **`bigpipe tping_svc`** command.

The external service checker executes as root. The BIG/ip Controller launches an external service checker using the following shell command:

```
<path> <node_ip> <port> [ <additional_argument> ... ]
```

For the case of the example shown above, the appropriate command would be:

```
/usr/local/lib/pingers/my_pinger n1 8000 -a 600 -b
```

The BIG/ip Controller inserts the node IP and port number before the additional arguments that are specified in the */etc/bigd.conf* file.

Note that the standard input and output of an external service checker are connected to Bigdnode. Bigdnode does not write anything to the external service checker's standard input, but it does read the external service checker's standard output. Whenever Bigdnode is able to read anything at all from the external service checker program, then that is treated as success, and the particular service is considered *up*.

8

Monitoring the BIG/ip Controller Using Command Line Utilities

- **Monitoring utilities provided on the BIG/ip platform**
- **Using the BIG/pipe command utility as a monitoring tool**
- **Working with the BIG/stat utility**
- **Working with the BIG/top utility**
- **Working with the Syslog utility**

Monitoring utilities provided on the BIG/ip platform

The BIG/ip platform provides several monitoring utilities for the command line. You can monitor system statistics, as well as statistics specific to virtual servers and nodes, such as the number of current connections, and the number of packets processed since the last reboot.

The BIG/ip platform provides the following monitoring utilities:

- **BIG/pipe**
  If you type certain BIG/pipe commands, such as **`bigpipe vip`** or **`bigpipe node`**, but do not include keywords in the command, the command displays statistical information about the elements that you configure using that command.
- **BIG/stat**
  This utility is provided specifically for statistical monitoring of virtual servers, nodes, NATs, and services. One benefit of using BIG/stat is that it allows you to customize the display of statistical information.
- **BIG/top**
  BIG/top provides real-time statistical monitoring. You can set a refresh interval, and you can specify a sort order.
- **Syslog**
  Syslog is the standard UNIX system logging utility, which monitors critical system events, as well as configuration changes made on the BIG/ip Controller.

Using the BIG/pipe command utility as a monitoring tool

Using the BIG/pipe utility, you can view information about the BIG/ip Controller itself, as well as elements such as virtual servers, virtual addresses, virtual ports, nodes, and node addresses. Typically, the BIG/ip Controller provides the following statistics:

- Current number of connections
- Total number of connections since the last system reboot
- Total number of bits (inbound, outbound, total)
- Total number of packets (inbound, outbound, total)

Monitoring the BIG/ip Controller

The **`bigpipe summary`** command displays performance statistics for the BIG/ip Controller itself. This display summary includes up-to-the-minute usage statistics, such as the amount of time a BIG/ip Controller has been running since the last reboot, or since the BIG/ip Controller became the active unit in a redundant system. The command syntax is simply:

`bigpipe summary`

The BIG/ip Controller displays the performance statistics in the format shown in Sample Screen 8.1 below.

```
BIGIP  total uptime            = #(day) #(hr) #(min) #(sec)
BIGIP  total uptime (secs)     =
BIGIP  total # connections     =
BIGIP  total # pkts            =
BIGIP  total # bits            =
BIGIP  total # pkts (inbound)  =
BIGIP  total # bits (inbound)  =
BIGIP  total # pkts (outbound) =
BIGIP  total # bits (outbound) =
BIGIP  current # connections   =
BIGIP  err.port_deny           =
BIGIP  err.no_nodes            =
BIGIP  err.reaper              =
```

Sample Screen 8.1 *The BIG/pipe summary display screen*

Table 8.1 describes the individual statistics included in the summary display screen.

Statistic	Description
`total uptime`	Total time elapsed since the BIG/ip Controller was last booted, or since the BIG/ip Controller became the active unit in a redundant system.
`total uptime (secs)`	Total uptime displayed in seconds.
`total # connections`	Total number of connections handled.
`total # pkts`	Total number of packets handled.
`total # bits`	Total number of bits handled.
`total # pkts (inbound)`	Total number of incoming packets handled.
`total # bits (inbound)`	Total number of incoming bits handled.
`total # pkts (outbound)`	Total number of outgoing packets handled.
`total # bits (outbound)`	Total number of outgoing bits handled.
`current # connections`	Total number of current connections.
`err.port_deny`	The number of times a client attempts connection to an unauthorized port (unauthorized port and source IP are logged via syslog).
`err.no_nodes`	The number of times the BIG/ip Controller has tried to make a connection to a node, but no nodes were available.
`err.reaper`	The number of connections reaped due to being idle.

Table 8.1 *BIG/pipe monitoring statistics*

Viewing the status of the interface cards

The **`bigpipe interface`** command displays the current status and the settings for both the external and internal interface cards. You can also use the **`bigpipe interface`** command to view information for a specific interface card, using the command syntax below:

`interface <ifname>`

Monitoring virtual servers, virtual addresses, and services

You can use different variations of the **`bigpipe vip`** command, as well as the **`bigpipe port`** command, to monitor information about virtual servers, virtual addresses, and services managed by the BIG/ip Controller.

Displaying information about virtual servers and virtual addresses

The **`bigpipe vip`** command displays the status of virtual servers (*up*, *down*, or *disabled*), the current number of connections to each virtual server, and the status of the member nodes that are included in each virtual server mapping. The status for individual member nodes includes whether the node is *up*, *down*, or *disabled*, and also includes the cumulative count of packets and bits received and sent by the node on behalf of the virtual server. The BIG/ip Controller displays the statistics as shown in Sample Screen Sample Screen 8.2 below.

```
bigpipe vip
VIP +------> 192.168.20.100
    |        (cur, max, limit, tot) = (0, 0, 0, 0)
    |        (pckts,bits) in = (0, 0), out = (0, 0)
    +---+--> PORT 23                 UP
        |    (cur, max, limit, tot) = (0, 0, 0, 0)
        |    (pckts,bits) in = (0, 0), out = (0, 0)
        NODE 192.168.103.30:23       UP
        |    (cur, max, limit, tot) = (0, 0, 0, 0)
        |    (pckts,bits) in = (0, 0), out = (0, 0)
        +--> PORT 21                 UP
        |    (cur, max, limit, tot) = (0, 0, 0, 0)
        |    (pckts,bits) in = (0, 0), out = (0, 0)
        NODE 192.168.103.30:21       UP
             (cur, max, limit, tot) = (0, 0, 0, 0)
             (pckts,bits) in = (0, 0), out = (0, 0)
```

Sample Screen 8.2 *Virtual server statistics screen*

If you want to view statistical information about one or more specific virtual servers, simply include the virtual servers in the **bigpipe vip** command as shown below:

```
bigpipe vip <virt addr>:<port>... <virt addr>:<port>
```

If you want to view statistical information about traffic going to one or more virtual addresses, specify only the virtual address information in the command:

```
bigpipe vip <virt addr>... <virt addr>
```

Displaying information about services

The **bigpipe port** command allows you to display information about specific virtual ports managed by the BIG/ip Controller. You can use the command to display information about all virtual services, or you can specify one or more particular virtual services.

To view information about all virtual services, use the following syntax:

```
bigpipe port
```

To view statistical information about one or more specific virtual services, simply include the service names or port numbers as shown below:

```
bigpipe port <port>... <port>
```

Monitoring nodes and node addresses

The **bigpipe node** command displays the status of all nodes configured on the BIG/ip Controller. The information includes whether or not the specified node is *up*, *down*, or *disabled*, and the number of cumulative packets and bits sent and received by each node on behalf of all virtual servers. The BIG/ip Controller displays the statistical information as shown in Sample Screen 8.3.

```
bigpipe node
|   NODE 192.168.103.20         UP
     |   (cur, max, limit, tot) = (0, 0, 0, 0)
     |   (pckts,bits) in = (0, 0), out = (0, 0)
     +---PORT 23            UP
         (cur, max, limit, tot) = (0, 0, 0, 0)
         (pckts,bits) in = (0, 0), out = (0, 0)
```

Sample Screen 8.3 *Node statistics screen*

If you want to view statistical information about one or more specific nodes, simply include the nodes in the **`bigpipe node`** command as shown below:

`bigpipe node <node addr>:<port>... <node addr>:<port>`

If you want to view statistical information about traffic going to one or more node addresses, specify only the node address information in the command:

`bigpipe vip <node addr>... <node addr>`

Working with the BIG/stat utility

BIG/stat is a utility that allows you to quickly view the status of the following elements:

- Virtual servers
- Services
- Nodes
- Network address translations (NATs)

The BIG/stat utility allows you to customize the statistics display. For example, you can customize your output to display statistics for a single element, or for selected elements. You can also have the display automatically updated at a user-specified time interval.

The **bigstat** command accepts one or more options, which allow you to customize the statistical display. When you use the **bigstat** command without specifying any options, the BIG/stat utility displays virtual servers, services, nodes, and NATs only one time.

The basic command syntax is:

```
bigstat [ options...]
```

Table Table 8.2 describes the options that you can use in the **bigstat** command.

Option	Description
`-bigip`	Displays totals for the BIG/ip Controller overall.
`-c <count>`	Sets the interval at which new information is displayed.
`-h and -help`	Displays the help options.
`-nat`	Displays network address table (NAT) entries only.
`-no_viptot`	Removes virtual server totals from the display.
`-no_nodetot`	Removes node totals from the display.
`-node`	Displays nodes only.
`-port`	Displays ports only.
`-v`	Displays version information.
`-vip`	Displays virtual servers only.

***Table 8.2** The **bigstat** command options*

Working with the BIG/top utility

BIG/top is a real-time statistics display utility. The display shows the date and time of the latest reboot and lists activity in bits, bytes, or packets. Similar to BIG/stat, the BIG/top utility accepts options which allow you to customize the display of information. For

example, you can set the interval at which the data is refreshed, and you can specify a sort order. The BIG/top displays the statistics as shown in Sample Screen 8.4 below.

```
                   |  bits  since       |  bits in prior     |   current
                   |  Nov 28 18:47:50   |  3 seconds         |   time
BIG/ip      ACTIVE |---In----Out---Conn-|---In----Out---Conn-|   00:31:59
227.19.162.82           1.1G  29.6G 145     1.6K     0      0

VIP ip:port        |---In----Out---Conn-|---In----Out---Conn-|-Nodes Up--
217.87.185.5:80       1.0G  27.4G 139.6K   1.6K     0      0       2
217.87.185.5:20      47.5M  2.1G    3.1K     0      0      0       2
217.87.185.5:20      10.2M 11.5M    2.6K     0      0      0       2

NODE  ip:port      |---In----Out---Conn-|---In----Out---Conn-|--State----
129.186.40.17:80    960.6M  27.4G  69.8K   672      0      0      UP
129.186.40.17:20     47.4M   2.1G   3.1K     0      0      0      UP
129.186.40.18:80    105.3M 189.0K  69.8K   1.0K     0      0      UP
129.186.40.17.21      9.4M  11.1M   1.3K     0      0      0      UP
129.186.40.18:21    700.8K 414.7K   1.3K     0      0      0      UP
129.186.40.18:20      352    320      1      0      0      0      UP
```

Sample Screen 8.4 *The BIG/top screen display*

Using BIG/top command options

The **bigtop** command uses the syntax below, and it supports the options outlined in Table 8.2:

```
bigtop  [options...]
```

Option	Description
-bytes	Displays counts in bytes (the default is bits).
-conn	Sorts by connection count (the default is to sort by byte count).
-delay <value>	Sets the interval at which data is refreshed (the default is 4 seconds).
-delta	Sorts by count since last sample (the default is to sort by total count).
-help	Displays BIG/top help.
-nodes <value>	Sets the number of nodes to print (the default is to print all nodes).
-nosort	Disables sorting.
-once	Prints the information once and exits.
-pkts	Displays the counts in packets (the default is bits).
-scroll	Disables full-screen mode.
-vips <value>	Sets the number of virtual servers to print (the default is to print all virtual servers).

Table 8.2 *BIG/top command options*

Using runtime commands in BIG/top

The BIG/top utility continually updates the display at the rate indicated by the **-delay** option. If you specified a value for this option, you can also use the following runtime options at any time:

- The **u** option cycles through the display modes; bits, bytes, and packets.
- The **q** option quits the BIG/top utility.

Working with the Syslog utility

The BIG/ip Controller supports logging via the Syslog utility. The logs are generated automatically, and saved into user-specified files. These logs contain all changes made to the BIG/ip Controller

configuration, such as those made with the **`bigpipe vip`** command, or other BIG/pipe commands, as well as all critical events that occur in the system.

Note

You can configure the Syslog utility to send email or activate pager notification based on the priority of the logged event. For more information about this, and other Syslog configuration issues, refer to Chapter 6.

The Syslog log files track system events based on information defined in the */etc/syslog.conf* file. You can view the log files in a standard text editor, or with the "less" file page utility.

Sample log messages

The following sample log messages give you an idea of how the Syslog utility tracks events that are specific to the BIG/ip Controller.

Sample message	Description
`bigd: allowing connections on port 20`	A user specifically allowed connections on virtual port 20
`bigd: node 192.168.1.1 detected up`	The 192.168.1.1 node address was successfully pinged by the BIG/ip Controller
`bigd: added service port 20 to node 192.168.1.1`	A user defined a new node, 192.168.1.1:20.
`bigd: security: port denial 207.17.112.254:4379 -> 192.168.1.1:23`	A client was denied access to a specific port. The client is identified as coming from 207.17.112.254:4379, and the destination node is 192.168.1.1:23.

Table 8.3 *Sample Syslog messages*

9

Load Balancing

- **Working with load balancing modes**
- **Setting a load balancing mode**
- **Working with persistence**

Working with load balancing modes

Load balancing is an integral part of the BIG/ip platform. A load balancing mode defines, in part, the logic that a BIG/ip Controller uses to determine which node should receive a connection hosted by a particular virtual server. The BIG/ip platform supports seven different load balancing modes, three of which are static modes, and four of which are dynamic modes. A static load balancing mode distributes connections based solely on user-defined settings, while a dynamic load balancing mode distributes connections based on various aspects of real-time server performance analysis.

Because each application of the BIG/ip Controller is unique, and server performance depends on a number of different factors, we recommend that you experiment with different load balancing modes, and choose the one that offers the best performance in your particular environment. For many sites, a static load balancing mode, such as Round Robin, achieves very acceptable results. Sites that have specific concerns, such as servers that vary significantly in speed and capability, may benefit from using dynamic load balancing modes.

The selected load balancing mode applies to all nodes configured on the BIG/ip Controller. In the BIG/config application, you set the load balancing mode on the BIG/ip System Properties page. If you want to configure load balancing or view the currently selected load balancing mode using the BIG/pipe command line utility, you use the **`bigpipe lb`** command.

Static load balancing modes

The BIG/ip platform supports three static load balancing modes: Round Robin, Ratio, and Priority. Each of these modes distributes connections in a specific order. Note that static load balancing modes do not take current node performance or load into account.

Round Robin mode

The Round Robin mode is the default load balancing mode. In Round Robin mode, the BIG/ip Controller distributes connections evenly across the nodes that it manages. Each time a new connection is requested, the BIG/ip Controller passes the connection to the next node in line.

Over time, the total number of connections received by each node associated with a specific virtual server is the same. Round Robin mode works well in environments where content servers have similar hardware capabilities, and there is a fairly even distribution of nodes to virtual servers for the entire BIG/ip system.

Ratio mode

The Ratio mode allows you to assign weights to each node. Over time, the total number of connections for each node is in proportion to the specified weights. For example, in simple configuration, you might have one new, fast server that hosts a node, and two older, slower servers that each host one node. If you were to use Ratio mode in this configuration, you would assign a higher ratio to the node on the fast server, and lower ratios to the nodes on the two slower servers. In a typical configuration, you might set the node on the fast server to receive twice as many connections as each of the nodes on the slow servers. Over time, the node on the fast server would receive 50% of the virtual server traffic, while each of the nodes on the slow servers would receive 25% of the virtual server traffic.

The default ratio for all nodes is 1. If you use the Ratio load balancing mode, you must change the ratio setting for at least one node; otherwise Ratio mode has the same result as Round Robin mode. Ratio mode works well in environments where one or more servers can handle significantly more connections than the other servers in the array.

Priority mode

In Priority mode, you create groups of nodes and assign a priority level to each group. The BIG/ip Controller distributes connections in a round robin fashion to all nodes in the highest priority group.

Should all the nodes in the highest priority group go down, the BIG/ip Controller begins to pass connections on to nodes in the next lower priority group. For example, in a configuration that has three priority groups, connections are first distributed to all nodes set as priority 1. If all priority 1 nodes are down, connections begin to be distributed to priority 2 nodes. If both the priority 1 nodes and the priority 2 nodes are down, connections then begin to be distributed to priority 3 nodes, and so on. Note, however, that the BIG/ip Controller continuously monitors the higher priority nodes, and each time a higher priority node becomes available, the BIG/ip Controller passes the next connection to that node.

Dynamic load balancing modes

The BIG/ip Controller supports four dynamic load balancing modes: Least Connections, Fastest, Observed, and Predictive. Dynamic load balancing modes distribute connections based on performance monitoring as well as current connection count, depending on the selected mode.

Least Connections mode

The Least Connection mode is relatively simple in that the BIG/ip Controller passes a new connection to the node with the least number of current connections. Least connections mode works best in environments where the servers in the array have similar capabilities.

Fastest mode

The Fastest mode passes a new connection based on the fastest response of all currently active nodes. Fastest mode works well in any environment, but may be particularly useful in environments where the nodes are hosted by physical servers of varying capabilities, or where nodes are distributed across different logical networks.

Observed mode

Observed mode is a combination of the logic used in the Least Connection and Fastest modes. In Observed mode, nodes are ranked based on a combination of the number of current connections and the response time. The node that has the best balance of fewest connections and fastest response time receives the next connection from the BIG/ip Controller. Observed mode also works well in any environment, but may be particularly useful in environments where node performance varies significantly.

Predictive mode

Predictive mode also uses the ranking methods used by Observed mode, where nodes are rated according to a combination of the number of current connections and the response time. However, in Predictive mode BIG/ip Controller analyzes the trend of the ranking over time, determining whether a node's performance is currently improving or declining. The node with the best performance ranking that is currently improving, rather than declining, receives the next connection from the BIG/ip Controller. Predictive mode works well in any environment.

Setting a load balancing mode

You can set the load balancing mode, or view the currently selected mode using the BIG/config application, or using the BIG/pipe command line utility.

Setting a load balancing mode in the BIG/config application

In the BIG/config application, the load balancing mode is set in the BIG/ip system properties. To view the BIG/ip system properties, click the BIG/ip Controller icon at the top of the System Tree. The **Load Balancing Method** box displays the currently selected load balancing mode. You can change the load balancing mode by selecting a new mode from the list box, and clicking the Apply

button. Note that your changes are not permanent unless you save the BIG/ip system configuration (see Save Configuration on the BIG/ip System Properties screen). If you do not save the configuration, the load balancing mode is reset the next time the BIG/ip system reboots either manually, or during a fail-over.

WARNING

If you change the load balancing mode to Ratio or Priority, you must also define ratio or priority settings for node addresses in the BIG/ip system configuration. Ratio and priority settings are defined in the node address properties, and the default setting is 1. To view properties for a particular node address, click ***Virtual Servers*** *in the System Tree, and then click the desired node address displayed on the Virtual Servers page.*

Setting a load balancing mode using the BIG/pipe command utility

The **`bigpipe lb`** command sets the load balancing mode for the BIG/ip Controller. The load balancing command syntax includes the **`<mode>`** parameter for which you have to specify the name of the load balancing mode you want to use:

```
bigpipe lb <mode>
```

Table 9.1 displays the command syntax for the BIG/pipe load balancing mode command.

Command	Description
`bigpipe lb`	Displays load balancing mode currently in use.
`bigpipe lb rr`	Sets load balancing to Round Robin mode.
`bigpipe lb ratio`	Sets load balancing to Ratio mode.
`bigpipe lb priority`	Sets load balancing to Priority mode.
`bigpipe lb least_conn`	Sets load balancing to Least Connections mode.
`bigpipe lb fastest`	Sets load balancing to Fastest mode.
`bigpipe lb observed`	Sets load balancing to Observed mode.
`bigpipe lb predictive`	Sets load balancing to Predictive mode.

Table 9.1 *Command syntax for setting load balancing mode*

WARNING

If you set the load balancing mode to Ratio or Priority, you must define the ratio or priority settings for each node address. The value you define using the **`bigpipe ratio`** *command is used as the ratio value if Ratio is the currently selected load balancing mode, and the same value is used as the priority level if Priority is the currently selected load balancing mode.*

Working with persistence

The BIG/ip Controller always has load balancing turned on; however, certain connections need to be sent to a specific node, rather than to a node selected by the load balancing algorithm. The BIG/ip Controller overrides the load balancing algorithm for connections that require persistence, such as those that deal with Active Server Pages, or e-commerce shopping carts.

Understanding persistence

The BIG/ip platform supports persistence for TCP, UDP, and SSL connections. When persistence is turned on for a specific service, clients can reconnect to a particular node in order to continue a previous session. For example, a client can establish an SSL connection to a travel site and reserve an airline ticket. The travel site may store the ticket information for a limited period of time, such as 20 minutes, and allow the client to reconnect to the site and purchase the ticket without having to re-enter the order. If the travel site stores the ticket information only on the node which hosted the original session rather than on a back-end database, the BIG/ip Controller's persistence setting overrides load balancing and sends the returning client to the original node.

When simple TCP persistence is enabled, the BIG/ip Controller actually records the IP address of the client, and it also records the particular node that received the initial client connection. When a new connection request comes from the same client, the BIG/ip Controller uses a look-up table to determine the appropriate node that should host the connection. The client record is cleared from the look-up table when the persistence timeout expires.

Note

For maximum performance, you should configure persistence timeout settings on the BIG/ip Controller so that they correlate to the amount of time that nodes typically retain the information that would be associated with a connection requiring persistence.

Persistence timeout settings

The BIG/ip platform supports two types of persistence timeout settings:

- The standard persistence timeout mode is the default timeout mode used on the BIG/ip Controller. A standard persistence timeout starts when a connection is first made and the timer runs until the timeout expires. The BIG/ip Controller sends subsequent connections to the same node until the timeout

expires. Once the timeout expires, however, the BIG/ip Controller treats a request for a subsequent connection as if it were new, and starts a new timeout period.

- The BIG/ip Controller offers an alternate persistence timeout mode where the timer resets itself upon receipt of each packet. Essentially, this keeps the timer from running as long as there is traffic flow over the connection. Once traffic stops on the connection, the timer runs as normal. Note that the timer is reset if traffic over the current connection resumes, or if the client subsequently reconnects before the timer actually expires.

Controlling the persistence timer

The persistence timeout mode is actually controlled by a persistence timeout system control variable. The default setting for this variable is enabled, which is the standard persistence timeout mode. In the BIG/config application, you can easily disable the variable by clearing the **Reset Persistence Timer On Each Packet** box in the BIG/ip Controller Advanced System Properties page.

You can also use the Sysctl command line utility to change this system control variable. To view the currently selected mode from the command line, type:

```
sysctl bigip.persist_time_used_as_limit
```

To activate this persistence mode, type:

```
sysctl -w bigip.persist_time_used_as_limit=1
```

To deactivate this persistence mode, type:

```
sysctl -w bigip.persist_time_used_as_limit=0
```

Maintaining persistence across all virtual servers

You can set the BIG/ip Controller to maintain persistence for all connections requested by the same client, regardless of which virtual server hosts each individual connection initiated by the client. When this mode is turned on, the BIG/ip Controller attempts to send all persistent connection requests received from the same client, within the persistence time limit, to the same node.

Connection requests from the client that do not use persistence are load balanced according to the currently selected load balancing mode.

For example, say a BIG/ip Controller configuration included the following virtual server mappings, where each virtual server uses persistence:

```
bigpipe vip v1:http define n1:http n2:http
bigpipe vip v1:ssl  define n1:ssl  n2:ssl
bigpipe vip v2:http define n1:http n2:http
bigpipe vip v2:ssl  define n1:ssl  n2:ssl
```

Say that a client makes an initial connection to **`v1:http`** and the BIG/ip Controller's load balancing mechanism chooses **`n1:http`** as the node. If the same client subsequently connects to **`v2:ssl`**, the BIG/ip Controller would send the client's request to **`n1:ssl`**, which uses the same node address as the **`n1:http`** node that currently hosts the client's initial connection.

WARNING

In order for this mode to be effective, virtual servers that use TCP or SSL persistence should include the same node addresses in the virtual server mappings.

The system control variable **`bigip.persist_on_any_vip`** turns this mode on and off. To activate the persistence mode, type:

```
sysctl -w bigip.persist_on_any_vip=1
```

To deactivate the persistence mode, type:

```
sysctl -w bigip.persist_on_any_vip=0
```

Maintaining persistence across virtual servers that use the same virtual addresses

The BIG/ip platform provides a similar persistence mode that is more granular. The BIG/ip Controller can maintain persistence for all connections requested by the same client, as long as the virtual server hosting each request uses the same virtual address. When this mode is turned on, the BIG/ip Controller attempts to send all persistent connection requests received from the same client, within

the persistence time limit, to the same node only when the virtual server hosting the connection has the same virtual address as the virtual server hosting the initial persistent connection. Connection requests from the client that go to other virtual servers with different virtual addresses, or those connection requests that do not use persistence, are load balanced according to the currently selected load balancing mode.

Using the preceding example, say a BIG/ip Controller configuration included the following virtual server mappings, where each virtual server uses persistence:

```
bigpipe vip v1:http define n1:http n2:http
bigpipe vip v1:ssl  define n1:ssl  n2:ssl
bigpipe vip v2:http define n1:http n2:http
bigpipe vip v2:ssl  define n1:ssl  n2:ssl
```

Say that a client makes an initial connection to **`v1:http`** and the BIG/ip Controller's load balancing mechanism chooses **`n1:http`** as the node. If the same client then connects to **`v2:ssl`**, the BIG/ip Controller starts tracking a new persistence session, and it uses the load balancing mode to determine which node should receive the connection request because the requested virtual server uses a different virtual address (**`v2`**) than the virtual server hosting the first persistent connection request (**`v1`**). However, if the client subsequently connects to **`v1:ssl`**, the BIG/ip Controller uses the persistence session established with the first connection to determine the node that should receive the connection request, rather than the load balancing mode. The BIG/ip Controller should send the third connection request to **`n1:ssl`**, which uses the same node address as the **`n1:http`** node that currently hosts the client's first connection with which it shares a persistent session.

WARNING

In order for this mode to be effective, virtual servers that use the same virtual address, as well as use TCP or SSL persistence, should include the same node addresses in the virtual server mappings.

The system control variable **`bigip.persist_on_any_port_same_vip`** turns this mode on and off. To activate the persistence mode, type:

```
sysctl -w bigip.persist_on_port_same_vip=1
```

To deactivate the persistence mode, type:

```
sysctl -w bigip.persist_on_port_same_vip=0
```

Configuring TCP and UDP persistence

You have to specifically enable TCP and UDP persistence for each virtual port that requires it. In the BIG/config application, you set TCP and UDP persistence timeouts in the Global Virtual Port Properties screen. Each virtual server using the virtual port inherently uses the persistence settings you define. Note that all persistence times are measured in seconds. The simple persistence used by TCP connections can be augmented for secure connections by setting SSL persistence on specific virtual servers.

You can also set persistence for TCP and UDP connections using the **`bigpipe persist`** command.

Configuring SSL persistence

You can set SSL persistence on those virtual servers for which you allow SSL connections. In the BIG/config Virtual Server Properties screen, you can allow SSL traffic for the virtual server and set the SSL persistence timeout. Note that you can also set a separate SSL timeout which applies to idle SSL connections.

To set SSL persistence for a virtual server using the BIG/pipe command line utility, you use the **`bigpipe vip`** command with a special parameter syntax:

```
bigpipe vip <virt addr:port> define <node addr:port> \
   <node addr:port> special <protocol> <persistence timeout> \
   <connection record timeout>
```

In the above command, **`<protocol>`** should be set to **`ssl`**, **`<persistence timeout>`** is the time allowed for a session to be subsequently re-established, and **`<connection record timeout>`** is the time allowed for the connection record to remain in the BIG/ip Controller's look-up table. For example, the following command sets SSL on virtual server **`v1`**, with a timeout of on hour, or 3600 seconds, and sets the BIG/ip Controller to keep the SSL session ID record information for two hours, or 7200 seconds.

```
bigpipe vip v1:ssl define n1:ssl n2:ssl special ssl 3600 7200
```

Note that the persistence timeout and the connection record timeout do not have to match. In fact, you may want to set the connect record timeout higher, because the BIG/ip Controller tracks the number of times a session ID match occurs where the persistence time for the session has actually expired. This statistic can help you to determine whether or not the current persistence setting is appropriate for your site traffic.

For diagnostic purposes, the bigpipe ss command displays the session IDs currently stored in the look-up table. The display includes each ID, timestamp, and the node address and node port which hosted the session. Note that the "Maximum hash table entries" is the highest number of hash table entries observed so far, not a limit.

Understanding SSL persistence

Each time a new SSL session starts, the receiving server must exchange an SSL handshake with the client, during which the client and server establish a session ID, exchange security certificates, and negotiate an encryption and compression method. When a client wishes to re-establish a connection with the server, the client identifies itself using the session ID created during the handshake of the original conversation. If the server accepts the request and restarts the previous session, the client bypasses the SSL handshake process and continues to use the same encryption methods.

One important aspect of SSL persistence is that it does not require that connections which are re-establishing a session go through the client authentication process. Authentication is resource intensive, and it reduces the overall throughput of the server. Persistence also improves server throughput because it allows the client to bypass the SSL handshake that would otherwise be required. This can provide significant performance improvement for protocols such as HTTP, which often involves opening connections to the same server several times to transfer a single web page.

SSL persistence and dynamic IP addresses

In some network configurations, the client's IP address may change from one connection to the next. Actually, networks are often configured to change the IP address of a client on a regular basis. Many firewalls, for example, translate the network addresses used by clients into one or more IP addresses that the firewall manages on behalf of the clients. In this way, the firewall directs traffic to and from the outside network without actually exposing the IP addresses used within its protected network. In addition to translating a client's IP address, a firewall may also translate the port number. By translating the port number, the firewall can use the same IP address among multiple clients (sometimes referred to as address overloading). Address overloading allows a network behind a firewall to make thousands of connections to the internet using only one IP address. However, in a large network, a single firewall may spread traffic across several IP addresses, or the network may use more than one firewall handling the traffic. Each firewall can use a different IP address for the traffic passing through it. In this case, a firewall or array of firewalls may translate a client's IP address into a different address for each TCP session.

In these types of configurations, simple persistence alone does not work. Because SSL persistence uses session IDs as the client identifier instead of a client IP address, SSL persistence is more appropriate.

Using SSL persistence with simple persistence

You may want to use SSL persistence and simple persistence together. In situations where the SSL persistence times out and the session information is discarded, or if a returning client does not provide a session ID, it may still be desirable for the BIG/ip Controller to direct the client to the original node using the IP address. The BIG/ip Controller can accomplish this as long as the client's simple persistence record is still in the BIG/ip Controller look-up table.

A

Glossary

Term	Definition
active unit	A BIG/ip Controller unit in a redundant system, which currently accepts and distributes connections. If the active unit in the redundant system fails, the standby unit takes over.
bandwidth	The transmission or processing capacity of a system or of a specific location or component in a system. A greater transmission rate can be achieved with a greater bandwidth.
BIG/ip Controller	Service Array Controller that monitors each server for application availability and performance, and automatically routes incoming queries to the most available server.
BIG3d	The listener which runs on each BIG/ip Controller and answers 3DNS system queries.
BIND (Berkley Internet Name Domain)	The most common implementation of DNS.
browser	A software program that retrieves, displays, and prints information and HTML documents from the WWW.
caching	Storing or buffering data in a temporary location so that the information can be retrieved quickly by an application.
content	Electronic information of value, including software, video, audio, and data. Dynamic content is content that is continually changing based on user interaction, such as multimedia games or database applications. Static content is content that does not change, such as an information page on a Web site.
daemon	A transport agent program that runs in the background on UNIX systems and responds to requests from users.
DNS (Domain Name System)	A distributed database that maps IP addresses to host names.
DNS server	See name server.

Term	Definition
domain name	The unique name that identifies an Internet site, such as www.f5.com. A given computer may have more than one domain name, but a given domain name points to only one computer.
ECV service check	One of three types of BIG/ip Controller service checks. ECV service check performs the service check using the extended content verification feature (see Extended Content Verification, and Service Check).
EAV service check	One of three types of BIG/ip Controller service checks. EAV service check executes an external service checker program, which actually performs the service check function on behalf of the BIG/ip Controller (see Extended Application Verification, external service checker program, and Service Check).
encryption key	The sequence of data that prevents unauthorized access to other data.
Ethernet 802.3	A protocol for networking computers in a LAN at speeds up to 10Mbps. Uses several varieties of physical medium dependent protocols including 10BASE5, 10BASE2, and 10BASET.
Extended Application Verification (EAV)	A BIG/ip feature that allows you to use an external program to determine a node's status based on whether the node returns specific content.
Extended Content Verification (ECV)	A BIG/ip feature that allows you to determine a node's status based on whether the node returns specific content.
external interface	The network interface on which the BIG/ip Controller receives connection requests. In a standard configuration, this is typically the external network where external clients request connections to internal servers. In a Transparent Node Mode configuration, this is typically the internal network where internal clients request connections to external servers.

Term	Definition
external service checker program	A custom program that performs a service check on behalf of the BIG/ip Controller.
fail-over	The process of a standby BIG/ip Controller unit in a redundant system taking over when a software failure or a hardware failure is detected on the active BIG/ip Controller.
fail-over cable	The cable that directly connects the two BIG/ip Controller units in a redundant system.
FDDI (Fiber Distributed Data Interface)	A multi-mode protocol for transmitting data on optical-fiber cables up to 100Mbps.
firewall	A system (software and/or hardware) that prevents external intrusion into a private enterprise system or network. Provides a security gateway for a private system connecting to a wide area public network such as the Internet.
F-Secure SSH	An encryption utility that allows secure shell connections to the BIG/ip controller.
FTP (File Transfer Protocol)	A utility program used to download or upload files between computers on a network.
gateway	Hardware and software that forward data between two networks.
hit	A successful access to a file on a Web page.
home page	The main HTML page seen by users at a WWW site.
host	Any computer on a network that makes services available to other computers on the network.
host machine	For the purposes of this manual, "host machine" refers to a single network server or other server array controller.
HTML (HyperText Markup Language)	A coding system used to format documents for viewing on the WWW.

Term	Definition
HTTP (HyperText Transfer Protocol)	An Internet computer communication encoding standard for the exchange of information and multimedia documents on the WWW.
HUP	A BIND name server signal. It restarts the name server. Use this signal after modifying the name server's boot file or one of its database files for the changes to take effect. You can also send this signal to BIND 4.93 secondary name server so as to update its secondary zones.
ICMP (Internet Control Message Protocol)	An Internet communications protocol. This protocol provides information relevant to IP packet processing and error correction.
INT	A BIND name server signal. It saves a copy of the name server's database to a file called *named_dump.db.* This file is located in */var/tmp* or */usr/tmp*, depending on your configuration.
internal interface	The network interface on which the BIG/ip Controller distributes connections. In a standard configuration, this is the network that houses the servers (nodes). In a Transparent Node Mode configuration, this is the network that houses the routers, or router-like devices.
Internet	The global network of networks that grew out of the Department of Defense funded research project.
InterNIC	An organization that registers domain names and IP addresses and distributes information about the Internet. InterNIC's Internet address is rs.internic.net.
intranet	Internal enterprise networks that use WWW products and services.
IP address	A unique number consisting of four parts separated by dots, e.g., 125.6.113.67 Every machine on the Internet has a unique Internet protocol address.

Term	Definition
iQuery	A UDP based protocol used to communicate and exchange information between BIG/ip Controllers and 3DNS systems.
ISP (Internet Service Provider)	A business that allows companies and individuals to connect to the Internet by providing the interface to the Internet backbone.
Java	An object-oriented programming language that allows Web pages to display applets (small programs that can create sound and animation).
LAN (Local Area Network)	A private computer network limited to an immediate area, such as a building or a floor of a building. Typically connected using coaxial cable, twisted pair or multi-mode fiber.
load balancing	The distribution of network traffic among many lines or servers to smooth or accelerate access demand.
member	When a node is included in a particular virtual server mapping, the node is said to be a member of that specific virtual server.
name server	A computer that can answer DNS queries. Name servers contain information about some part of the DNS, and they make that information available to clients. Also called DNS server.
named (name server daemon)	The Internet domain name server.
network address translation (NAT)	An IP alias address that identifies servers managed by the BIG/ip Controller to the external network. You can define one NAT for each node address managed by the BIG/ip Controller.
node	A specific combination of an IP address and port number associated with a server in the array managed by the BIG/ip Controller.

Term	Definition
node address	The IP address associated with a node. This IP address can be the real IP address of a network server, or it can be an alias IP address on a network server.
node ping	A BIG/ip Controller function where the BIG/ip Controller issues standard echo pings to servers that host nodes in order to determine node status.
node port	The port number associated with a specific node.
node status	Whether a node is *up* and available to receive connections, or *down* and unavailable. The BIG/ip Controller uses node ping and service check to determine node status.
path	A route from a BIG/ip Controller to a local DNS.
persistence	A series of related connections received from the same client, having the same session ID. If persistence is turned on, the BIG/ip Controller sends all persistent connections to the same node.
port	A number that identifies a specific service offered by a host.
primary DNS	The machine that handles DNS name resolution.
resource record	The building blocks of the DNS. A resource record (RR) consists of a name, a type, and data that is specific to the type. These resource records, in a hierarchical structure, comprise the DNS
router	A router provides connectivity between enterprise networks (e.g., LANs and WANs), by forwarding information between networks using global addresses. Routers use special protocols to determine connectivity, and maintain addressing information.
server	Any computer that allows other computers to connect to it.
server application	A software application that runs on a server and manages user sessions.

Term	Definition
service check	A BIG/ip Controller function where the BIG/ip Controller either attempts to connect to the service hosted by the node, or issues an extended content verification request, or executes an external service checker program. The node response to any one of these service checks is used to determine the node's status.
SMTP (Simple Mail Transport Protocol)	The Internet standard protocol for the exchange of e-mail messages.
sod (switch over daemon)	A daemon that monitors, detects, and directs the fail-over process.
standby unit	A BIG/ip Controller unit in a redundant system, which is always prepared to become the active unit should the active unit fail.
TCP/IP (Transport Control Protocol/Internet Protocol)	A commonly used protocol suite for communicating across networks.
Telnet	A software service packaged with most operating systems that allows a user to log onto a computer over a network in the same way as if he or she were using a terminal attached to the computer.
Transparent Node Mode	A mode in which the BIG/ip Controller can perform load balancing on routers and router-like devices.
transparent node	A node that appears to other network devices, including the BIG/ip Controller, as a router.
URL (Uniform Resource Locator)	The URL provides information on the protocol, the system, and the file name so that the users system can find a particular document on the Internet.
virtual address	An IP address associated with a virtual server managed by the BIG/ip Controller.
virtual port	One component of a virtual server. The virtual port number should be the same TCP or UDP port number that is known to client programs.

Term	Definition
virtual server	A specific combination of a virtual address and virtual port, associated with a content site that is managed by a BIG/ip Controller or other host machine.
virtual server mapping	The list of nodes that accept connections made to the virtual server.
WAN (Wide Area Network)	Any intranet or network that covers an area larger than a single building or campus.
Web page	An HTML document on the World Wide Web.
Web server	A system capable of continuous access to the Internet through retrieving and displaying documents via HTTP.
Web site	The virtual location for an organization's presence on the WWW, usually made up of several Web pages and a single home page designated by a unique URL.
wildcard virtual server	A virtual server capable of accepting connections with a destination address outside the network (used only when the BIG/ip Controller runs Transparent Node Mode).
WKS (Well-Known Services)	A type of resource record that describes the services usually provided by a particular protocol on a particular port.
WWW (World Wide Web)	The mechanism to share documents via the Internet. The WWW allows computer users to access information across systems around the world using URLs to identify files and systems and hypertext links to move between files on the same or different systems.

B

BIG/pipe Command Reference

BIG/pipe commands

Command	Description	Page
alias	Defines an IP alias to be pinged on behalf of a specific group of nodes.	-4
configsync	Synchronizes the */etc/bigip.conf* between the two BIG/ip Controller units in a redundant system.	-6
-d	Parses the command line options without executing them.	-7
dt	Prints the current connection table.	-8
-f	Loads a specific configuration file.	-9
fo	Switches the BIG/ip Controller between active and standby in a redundant configuration.	-10
-h and -help	Displays online help.	-12
interface	Sets options on individual interfaces for redundant configurations.	-13
lb	Sets load balancing mode.	-17
maint	Toggles BIG/ip Controller into and out of maintenance mode.	-18
nat	Defines network address translations.	-19
node	Defines node property settings.	-22
persist	Defines settings for TCP persistence.	-25
port	Defines settings for virtual ports.	-27
ratio	Sets load-balancing weights and priority levels used in the Ratio and Priority load balancing modes.	-29
-s	Saves a specific configuration file.	-31
summary	Displays a summary of BIG/ip Controller usage statistics.	-32
timeout_node	Sets the amount of time node addresses have to respond to a ping issued by the BIG/ip Controller.	-34
timeout_svc	Sets the amount of time services have to respond to a service check issued by the BIG/ip Controller.	-36
tping_node	Sets the interval at which the BIG/ip Controller pings node addresses to determine node status.	-38
tping_svc	Sets the interval at which the BIG/ip Controller issues service checks to nodes to determine node status.	-40

Command	Description	Page
`treaper`	Sets the expiration time for idle connections on virtual ports.	-42
`udp`	Enables UDP on virtual ports, and sets UDP persistence settings.	-44
`-v`	Displays the BIG/pipe command version number.	-46
`version`	Displays the BIG/ip Controller OS version number.	-47
`vip`	Defines virtual servers, virtual server mappings, and virtual server properties.	-48

alias

Description

This command defines a representative node that is used to represent a group of node addresses that are actually IP aliases on the same physical server. To determine if the nodes associated with the representative node alias are available, the BIG/ip Controller sends a single node ping to the node alias, rather than an individual ping to each node address. If the BIG/ip Controller receives a response to the node alias ping, it marks the group of nodes, as *up* and available for connections. The command is useful only for large configurations that include 1,000 or more nodes.

Note that this is also effective for nodes that are configured for service check, as long as each node uses the same port number. Although the BIG/ip Controller performs the service check on the node alias, it opens the specific port that is associated with the node. If the BIG/ip Controller receives a valid response to the service check, it marks each node in the group *up*, assuming that the specific service is available on each node in the group.

Syntax

```
bigpipe alias
bigpipe alias <node>
bigpipe alias <node addr>... <node addr> pingnode <pingnode_ip>
```

Displaying current node aliases

The following command displays all node aliases defined on the BIG/ip Controller:

```
bigpipe alias
```

The following command displays the node alias defined for a specific node, where **<node>** is the node address and node port number:

```
bigpipe alias <node>
```

Defining a node alias

The following command defines the node alias for one or more node addresses, where **<pingnode_ip>** is the node alias (the node address that represents the group):

```
bigpipe alias <node addr>... <node addr> pingnode <pingnode_ip>
```

Example

The following command defines a node alias for two node addresses, 192.168.42.2 and 192.168.42.3. The BIG/ip Controller performs node pings and service checks on 192.168.42.1 to determine the availability of 192.168.42.2 and 192.168.42.3.

```
bigpipe alias 192.168.42.2 192.168.42.3 pingnode 192.168.42.1
```

Note

*The address that servers as the node alias (<**pingnode_ip**>) must be a node address that is already defined in one or more virtual server mappings.*

configsync

Description

This command is called after one or more BIG/pipe commands have changed the BIG/ip Controller configuration. The command downloads the entire configuration and writes it to the */etc/bigip.conf* file. If you have a BIG/ip redundant system and SSH RSA Authentication is set up between the two BIG/ip units, the **configsync** command also copies */etc/bigip.conf* on the local BIG/ip Controller to */etc/bigip.conf* on the remote BIG/ip Controller, and then loads the new configuration file on the remote BIG/ip Controller. (For more information on synchronizing configurations in redundant systems, refer to Chapter 5.)

You can use the configsync command as a shortcut for the following commands as long as the */etc/bigip.failover* file contains the remote BIG/ip Controller address:

```
bigpipe -s /etc/bigip.conf
scp /etc/bigip.conf root@<ip-address>:/etc/bigip.conf
ssh -l root <ip-address> /sbin/bigpipe -f /etc/bigip.conf
```

Syntax

```
bigpipe configsync
```

-d

Description

Parses the command line options without executing them.

This distinguishes between valid and invalid commands, and is particularly useful with the **-f** option, to validate the configuration file.

Syntax

bigpipe -d -f <filename>

Reads the specified file name and checks the syntax, without actually changing the configuration.

To read from standard input, you can use the hyphen character ("**-**") in place of the file name.

Example

bigpipe -d -f /etc/bigip.conf

dt

Description

This command prints the list of current connections, including client, virtual server, and node addresses. If the BIG/ip Controller is running Transparent Node Mode, the **bigpipe dt** command also prints the final destination address for connections.

Syntax

```
bigpipe dt
```

-f

Description

Runs a script file using the **`bigpipe -f <filename>`** option.

BIG/pipe commands, once executed, remain in memory. Whenever a BIG/ip Controller is powered down and rebooted, you must re-issue the BIG/pipe commands. To make this process easy, we recommend that you create a configuration file which contains the specific BIG/pipe commands you use to configure the BIG/ip Controller. If a BIG/ip Controller loses power and reboots, you can re-configure it by running the configuration file instead of retyping the BIG/pipe commands.

Example

```
bigpipe -f /etc/bigip.conf
```

Syntax

To run a configuration file, type:

```
bigpipe -f [<filename> | - ]
```

The **`<filename>`** parameter is the name of the configuration file containing the BIG/pipe commands. If you use a hyphen character ("**`-`**") in place of the **`<filename>`** parameter, or if you omit the parameter altogether, the BIG/ip Controller uses the standard input. The BIG/ip Controller also automatically loads the script file */etc/bigip.conf* at boot up. You should use this file to store your configuration. You create this file using the **`bigpipe -s`** command.

Note

The bigpipe **`-f`** *command resets all of the BIG/ip Controller settings before it loads settings from /etc/bigip.conf.*

fo

Description

Switches the BIG/ip Controller to be the active or the standby unit in a redundant configuration. This command should be used with care, and is provided only for special situations. The BIG/ip Controller automatically switches between active and standby modes, without operator intervention.

Example

```
bigpipe fo slave
```

The above example sets the BIG/ip Controller to be the standby unit in the redundant system.

Syntax

```
bigpipe fo
bigpipe fo master
bigpipe fo slave
```

Displaying the current mode

Use the following syntax to display the current mode in which the BIG/ip Controller is running:

```
bigpipe fo
```

Switching the current mode

Before you switch the current mode, first determine which mode the BIG/ip Controller is running using the command above. To switch the BIG/ip Controller to be the active unit, use the following syntax:

```
bigpipe fo master
```

To switch the BIG/ip Controller to be the standby unit, use the following syntax:

bigpipe fo slave

WARNING

Do not switch both machines in a redundant system to be the standby machine at the same time. Neither BIG/ip Controller accepts connections in this state, and your redundant system is effectively removed from network service.

-h and -help

Description

Accesses help for the BIG/pipe utility.

Syntax

An online help command is available when you enter any of the following commands:

```
bigpipe
bigpipe -h
bigpipe -help
```

interface

Description

This command sets the amount of time before fail-over is triggered in a BIG/ip redundant system, toggles the interface into and out of fail-safe mode, and sets the MAC address.

Note

The interface command may be used to change more than one toggle or setting at a time.

Example

The following sample **interface** commands incorporate three different network interface cards:

- exp0 - Intel NIC
- fpa0 - FDDI NIC
- de0 - DEC/SMC NIC

```
bigpipe interface exp0 failsafe arm
bigpipe interface fpa0 failsafe disarm
bigpipe interface de0 timeout 10
```

Note

Use the **ifconfig -a** *command to list the names of the currently installed interfaces.*

Syntax

```
bigpipe interface
bigpipe interface <ifname> [ internal | external ]
bigpipe interface <ifname> failsafe [ arm | disarm ]
bigpipe interface <ifname> [ timeout <seconds> ]
bigpipe interface <ifname> [ mac_masq <mac_addr ]
```

Specifying the internal or external interface

Use the following command syntax when specifying the internal or the external interface.

```
bigpipe interface <ifname> [ internal | external ]
```

Viewing the timeout setting

Use the following syntax to view the fail-over timeout setting for a specific interface

```
bigpipe interface <ifname> timeout
```

Displaying status for interfaces

Use the following syntax to display the current status and the settings for both the internal and the external interfaces.

```
bigpipe interface
```

Use the following syntax to display the current status and the setting for a specific interface.

```
bigpipe interface <ifname>
```

Setting the fail-safe timeout

Use the following syntax to set the amount of time, in seconds, that a router or a node has to respond to a BIG/ip Controller ARP request in order to be designated operational. Note that the default is 30 seconds.

```
bigpipe interface <ifname> timeout <seconds>
```

If the router or node fails to respond within the specified time, the BIG/ip Controller assumes the router or the external network interface is down, or that the node or the internal interface is down.

Warning messages are generated after half of the specified timeout period. In the case of an armed BIG/ip Controller in a BIG/ip redundant system, traffic is switched from the active unit to the standby unit at the end of the timeout period. Note that the fail-safe timeout is used only if the fail-safe option is armed on the interface.

Displaying the current fail-safe status

Use the following syntax to display the current status and settings for the BIG/ip Controller fail-safe mode.

```
bigpipe interface <ifname> failsafe
```

Arming and disarming the fail-safe mode

Use the following syntax to activate the BIG/ip Controller fail-safe mode.

```
bigpipe interface <ifname> failsafe arm
```

When armed, the active unit automatically switches to the standby unit whenever the active unit detects a failure of the specified network interface or the router. The default fail-safe mode is set to disarm.

You should arm the fail-safe mode only after you configure the BIG/ip Controller, and both the active and standby units are ready to be placed into a production environment.

WARNING

You must specify a default route before using the **`bigpipe interface failsafe`** *command. You specify the default route in the /etc/hosts and /etc/netstart files.*

Use the following syntax to deactivate the BIG/ip Controller fail-safe mode.

```
bigpipe interface <ifname> failsafe disarm
```

Setting the MAC address

Use the following syntax to set the MAC address that will be shared by both BIG/ip Controller units in the redundant system.

```
bigpipe interface <ifname> mac_masq <MAC addr>
```

The sharing of the MAC address allows for the use of the BIG/ip Controller in a network topology utilizing secure hubs.

The MAC address is determined by executing the **/sbin/ifconfig -a** command. Find the MAC address on both the active and standby, units and choose one that is similar but unique.

WARNING

You must specify a default route before using the **mac_masq** *command.*

For **mac_masq** changes to take effect, you must save your configuration and reboot the BIG/ip Controller.

Example

Suppose you want to set up **mac_masq** on the external interfaces. Using the **ifconfig -a** command on the active and standby units, you note that their MAC addresses are:

```
Active: exp0 = 0:0:0:ac:4c:a2
Standby: exp0 = 0:0:0:ad:4d:f3
```

In order to avoid collisions, you now must choose a unique MAC address. The safest way to do this is to select one of the addresses and logically **OR** the first byte with **0x40**. This makes the MAC address a locally administered MAC address.

In this example, either 40:0:0:ac:4c:a2 or 40:0:0:ad:4d:f3 would be suitable shared MAC addresses to use on both BIG/ip Controllers in a redundant system.

The shared MAC address is used only when the BIG/ip Controller is in active mode. When the unit is in standby mode, the original MAC address of the network card is used. On startup, or when transitioning from standby mode to active mode, the BIG/ip Controller sends gratuitous ARP requests to notify the default router and other machines on the local Ethernet segment that its MAC address has changed. See RFC 826 for more details on ARP.

Note

You can use the same technique to configure an internal interface MAC address.

lb

Description

Specifies a load balancing mode.

Example

The command below sets the load balancing mode to Least Connections, which routes new connects to the node which currently maintains the least number of connections.

```
bigpipe lb least_conn
```

Syntax

```
bigpipe lb
bigpipe lb rr
bigpipe lb round_robin
bigpipe lb ratio
bigpipe lb priority
bigpipe lb fastest
bigpipe lb least_conn
bigpipe lb predictive
bigpipe lb observed
```

Viewing the currently selected load balancing mode

Use the following syntax to display the currently selected load balancing mode.

```
bigpipe lb
```

Setting the load balancing mode

Use the following syntax to set the load balancing mode.

```
bigpipe lb <mode name>
```

The mode names allowed are displayed in the Syntax section above.

maint

Description

Toggles a BIG/ip Controller into and out of Maintenance mode. When in Maintenance mode, a BIG/ip Controller accepts no new connections, but it does allow existing connections to complete.

Example

bigpipe maint

The **maint** command interactively prompts the BIG/ip Controller to enter or exit the maintenance mode.

Syntax

bigpipe maint

If BIG/ip Controller is already in maintenance mode, the **maint** command takes the BIG/ip Controller out of maintenance mode. If the BIG/ip Controller has not been in maintenance mode for more than 20 minutes, the BIG/ip Controller immediately begins to accept new connections to its VIPs.

If BIG/ip Controller has been in maintenance mode for more than 20 minutes, all network ARP caches are automatically updated by the BIG/ip Controller; this process normally takes a few seconds. However, you can speed the process up by reloading the configuration file, as follows:

bigpipe -f /etc/bigip.conf

nat

Description

A network address translation (NAT) command defines a mapping between the IP address of a server behind the BIG/ip Controller and an unused address on the network in front of the BIG/ip Controller.

The primary reason to define a NAT is to allow one of the servers in the server array behind the BIG/ip Controller to initiate communication with a computer in front of or external to the BIG/ip Controller. A NAT allows a server to initiate, for example, a Telnet connection, an HTTP request, or DNS request to an IP address that is external to the BIG/ip Controller. A packet going from the server to an external IP address has its source address changed from the actual IP address (defined by **<internal_ip>**) to a virtual IP address (defined by **<external_ip>**). A packet going from an external IP address to the server has its destination address changed from the virtual IP address (defined by **<external_ip>**) to the actual IP address (defined by **<internal_ip>**). You should always use the actual IP address of an internal host as the **<internal_ip>**.

Example

```
bigpipe nat 11.0.0.100 11.0.0.101
```

Syntax

```
bigpipe nat
bigpipe nat <internal_ip>... <internal_ip>
bigpipe nat <internal_ip> to <external_ip> [ netmask \
       <netmask>[broadcast < broadcast_ip>] | /<bitmask>]
bigpipe nat <internal_ip>... <internal_ip> delete
```

Displaying status of NATs

Use the following syntax to display the status of all NATs included in the configuration:

```
bigpipe nat
```

Use the following syntax to display the status of one or more selected NATs:

```
bigpipe nat <internal_ip>... <internal_ip>
```

Defining a NAT

Use the following syntax to define a NAT.

```
bigpipe nat <internal_ip> to <external_ip> [ netmask \
       <netmask>[broadcast < broadcast_ip>] | /<bitmask>]
```

The node behind BIG/ip Controller with the IP address specified by **<internal_ip>** has a presence in front of the BIG/ip Controller as IP address **<external_ip>**. The netmask is optional.

Examples

```
bigpipe nat 11.0.0.100 to 10.0.140.100
bigpipe nat 11.0.0.100 to 10.0.140.100 netmask 255.255.255.0
bigpipe nat 11.0.0.100 to 10.0.140.100/24
```

Deleting NATs

Use the following syntax to delete one or more NATs from the system:

```
bigpipe nat <internal_ip>... <internal_ip> delete
```

Additional Restrictions

The **nat** command has the following additional restrictions:

- A virtual server cannot use the IP address defined in the **<external_ip>** parameter.
- A NAT cannot use a BIG/ip Controller's IP address.
- The IP address defined in the **<internal_ip>** parameter must be routable to a specific server behind the BIG/ip Controller.
- A NAT cannot use an internal or external IP address defined for and used by another NAT.
- You must delete a NAT before you can redefine it.

Error Checking

When you issue a nat command that results in an error, you may see the following behavior:

- Incorrect command syntax generates a BIG/pipe error message.
- If you issue a command that ignores the restrictions listed above the BIG/ip Controller generates a log message.

In either case, the **`nat`** command does not execute.

node

Description

Reads information about nodes and sets connection limits for nodes, and node addresses.

Example

bigpipe node 192.168.200.50:20

When you issue the above command, the BIG/ip Controller displays the following information for the specified node.

```
bigpipe node 192.168.200.50:20
NODE 192.168.200.50    UP
|    (cur, max, limit, tot) = (0, 0, 0, 0)
|    (pckts,bits) in = (0, 0), out = (0, 0)
+-   PORT 20           UP
     (cur, max, limit, tot) = (0, 0, 0, 0)
     (pckts,bits) in = (0, 0), out = (0, 0)
```

Sample Screen 2.1 Node status and statistics

Syntax

bigpipe node
bigpipe node <node addr>
bigpipe node <node addr>... <node addr>
bigpipe node <node addr>:<port>
bigpipe node <node addr>:<port>... <node addr><port>
bigpipe node <node addr> limit <limit>
bigpipe node <node addr>... <node addr> limit <limit>
bigpipe node <node addr>:<port>... <node addr>:<port> limit <limit>

Displaying status of all nodes

Use the following syntax to display status and statistical information for all nodes included in the configuration:

bigpipe node

The command reads each node's *up/down* status, the number of current connections, total connections, and connections allowed, and the number of cumulative packets and bits sent and received. The display format is as follows:

```
+- Node IP address    up/down status
   (cur, max, limit, tot) = 0,0,0,0
   (pckts,bits) in = (#, #), out = (#, #)
```

Displaying the status of nodes addresses

Use the following syntax to display status and statistical information for one or more node addresses:

bigpipe node <node addr>... <node addr>

The command reads each node address' *up/down* status, the number of current connections, total connections, and connections allowed, and the number of cumulative packets and bits sent and received.

Displaying the status of specific nodes

Use the following syntax to display status and statistical information for one or more specific nodes:

bigpipe node <node addr>:<port>...<node addr>:<port>

Setting connection limits for nodes

Use the following syntax to set the maximum number of connections allowed for one or more nodes:

bigpipe node <node addr>:<port>... <node addr>:<port> limit <limit>

Note that to remove a connection limit, you also issue the above command, but you set the **<limit>** variable to zero.

Setting connection limits for node addresses

Use the following syntax to set the maximum number of connections allowed for one or more node addresses:

bigpipe node <node addr>... <node addr> limit <limit>

Note that to remove a connection limit, you also issue the above command, but you set the **<limit>** variable to zero.

persist

Description

Enables TCP persistence on one or more virtual ports. Persistence forces new connections that have the same source address and port as a prior connection to use the same node and port as used by the prior connection for the specified period.

By default, persistence is disabled on all ports. Note that persistence is affected by certain system control variables. For more information, refer to Appendix C.

Syntax

```
bigpipe persist
bigpipe persist <port>
bigpipe persist <port> <seconds>
bigpipe persist <port> 0
```

Displaying persistence settings for virtual ports

Use the following syntax to display the number of seconds for which the BIG/ip Controller maintains persistence information for all virtual ports that have persistence turned on:

```
bigpipe persist
```

Use the following syntax to display persistence settings for a specific virtual port:

```
bigpipe persist <port>
```

Setting a persistence timeout

Use the following syntax to set the number of seconds for which the BIG/ip Controller maintains persistence information on a specific virtual port:

```
bigpipe persist <port> <seconds>
```

Turning persistence off

Use the following syntax to turn persistence off for a specific virtual port:

```
bigpipe persist <port> 0
```

port

Description

This command allows and denies traffic on virtual ports, and it also allows you to set connection limits on ports. You can use standard port numbers or standard port names (for example, *www* or *80*) in the command parameters. Note that the settings you define using this command affect all virtual servers that use the specific port.

The default is for all ports to be disabled.

A port is any valid port number, between 1 and 65535, inclusive, or any valid service name in the */etc/services* file.

Example

The following example uses both names and port numbers:

```
bigpipe port 23 allow
bigpipe port
```

The following output is then displayed:

```
PORT 23     telnet
(cur, max, limit, tot, reaped) = (0, 0, 0, 0, 0)
(pckts,bits) in = (0, 0), out = (0, 0)
```

Syntax

```
bigpipe port
bigpipe port <port>
bigpipe port <port>... <port>
bigpipe port <port>... <port> allow
bigpipe port <port>... <port> deny
bigpipe port <port>... <port> limit <limit>
bigpipe port <port>... <port> limit 0
```

Displaying the status of all virtual ports

Use the following syntax to display the status of virtual ports included in the configuration:

```
bigpipe port
```

Displaying the status for specific virtual ports

Use the following syntax to display the status of one or more virtual ports:

```
bigpipe port <port>... <port>
```

Allowing and denying virtual ports

You can allow or deny traffic to specific virtual ports. The default setting for all virtual ports is "denied." Use the following syntax to allow one or more virtual ports:

```
bigpipe port <port>... <port> allow
```

To deny access to one or more virtual ports:

```
bigpipe port <port>... <port> deny
```

Setting connection limits on ports

Use the following syntax to set the maximum number of connections allowed on a virtual port at one time. Note that you can configure this setting for one or more virtual ports.

```
bigpipe port <port>... <port> limit <limit>
```

To turn off a connection limit for one or more ports, use the above command, setting the **<limit>** parameter to zero.

```
bigpipe port <port>... <port> limit 0
```

ratio

Description

This command provides two functions related to load balancing:

- For the Ratio load balancing mode, the command sets the weight or proportions for one or more node addresses.
- For the Priority load balancing mode, the command sets the priority level. Note that multiple node addresses can have the same priority level setting.

Example

The following command sets the ratio to **3** for the specific node address:

```
bigpipe ratio 192.168.103.20 3
```

The following command displays the current ratio settings for all node address that have ratio settings.

```
bigpipe ratio
```

The following output is displayed:

```
192.168.200.51    ratio = 3
192.168.200.52    ratio = 1
```

Syntax

```
bigpipe ratio
bigpipe ratio <node addr>
bigpipe ratio <node addr> <weight>
bigpipe ratio <node addr>... <node addr>
bigpipe ratio <node addr>... <node addr> <weight>
```

Displaying the ratio settings for all node addresses

Use the following syntax to display the current ratio settings for all node addresses included in the configuration:

```
bigpipe ratio
```

Displaying the ratio settings for specific node addresses

Use the following syntax to display the ratio setting for one or more node addresses:

```
bigpipe ratio <node addr>... <node addr>
```

Setting a ratio for one or more node addresses

The default ratio setting for any node address is 1. If you use the Ratio or Priority load balancing modes, you must set a ratio other than 1 for at least one node address in the configuration. If you do not change at least one ratio setting, the load balancing modes have the same affect as the Round Robin load balancing mode.

Use the following syntax to set the ratio for one or more node addresses:

```
bigpipe ratio <node addr>... <node addr> <weight>
```

Note

The **<weight>** *parameter must be a whole number, greater than or equal to 1.*

-s

Description

Saves the current BIG/ip Controller configuration settings to a file. The default file name is */etc/bigip.conf*, but you can use alternate file names if desired.

Example

The following command saves the configuration to the */etc/bipip.conf* file.

```
bigpipe -s /etc/bigip.conf
```

Syntax

```
bigpipe -s [ <filename> | - ]
```

The **<filename>** parameter is the name of the file where the configuration is written. This configuration file may be used with the **bigpipe -f** command. If you do not specify a file name, or if you use a hyphen character ("**-**") in place of the **<filename>** parameter, the configuration is written to the standard output.

In order for configuration changes to take effect upon boot-up, you need to use BIG/pipe to save a default configuration file, for example **bigpipe -s /etc/bigip.conf**.

summary

Description

Displays a summary of up-to-the-minute usage statistics, including the total number and number of current connections made, the total number of bits transferred, and the amount of time the BIG/ip Controller has been running as the active unit.

Syntax

bigpipe summary

Output

The output display format for the **summary** command is shown in Sample Screen 2.2.

```
BIGIP total uptime  = #(day) #(hr) #(min) #(sec)
BIGIP total uptime (secs)        =
BIGIP total # connections        =
BIGIP total # pkts               =
BIGIP total # bits               =
BIGIP total # pkts (inbound)     =
BIGIP total # bits (inbound)     =
BIGIP total # pkts (outbound)    =
BIGIP total # bits (outbound)    =

BIGIP current # connections      =
BIGIP err.port_deny              =
BIGIP err.no_node                =
BIGIP err.reaper                 =
```

Sample Screen 2.2 Summary output display

Table 2.1 outlines each statistic displayed by the **summary** command.

Statistic	Description
`total uptime`	Total time elapsed since the BIG/ip Controller was last booted, or since the BIG/ip Controller became the active unit in a redundant system.
`total uptime (secs)`	Total uptime displayed in seconds.
`total # connections`	Total number of connections handled.
`total # pkts`	Total number of packets handled.
`total # bits`	Total number of bits handled.
`total # pkts (inbound)`	Total number of incoming packets handled.
`total # bits (inbound)`	Total number of incoming bits handled.
`total # pkts (outbound)`	Total number of outgoing packets handled.
`total # bits (outbound)`	Total number of outgoing bits handled.
`current # connections`	Total number of current connections.
`err.port_deny`	The number of times a client attempts connection to an unauthorized port (unauthorized port and source IP are logged via syslog).
`err.no_nodes`	The number of times the BIG/ip Controller has tried to make a connection to a node, but no nodes were available.
`err.reaper`	The number of connections reaped due to being idle.

Table 2.1 *Statistics monitored by the* **`summary`** *command*

timeout_node

Description

Sets the amount of time that a server has to respond to a BIG/ip Controller ping in order for the nodes hosted by the server to be marked *up*. If a server fails to respond within the specified time, the BIG/ip Controller assumes that the server is down, and the BIG/ip Controller no longer sends requests to the nodes hosted by the server. If the server responds to the next ping, or to subsequent pings, the BIG/ip Controller then marks node hosted by the server *up*, and resumes sending requests to those nodes.

The **timeout_node** default is 15 seconds.

Example

The sample command below sets the time-out to 33 seconds.

```
bigpipe timeout_node 33
bigpipe timeout_node
```

The following output is then displayed:

```
timeout_node 33
```

Syntax

```
bigpipe timeout_node
bigpipe timeout_node <seconds>
bigpipe timeout_node 0
```

Displaying the current timeout value

Use the following syntax to display the current timeout setting for node ping:

```
bigpipe timeout_node
```

Setting a timeout value for node ping

Use the following syntax to set the timeout setting for node ping:

```
bigpipe timeout_node <seconds>
```

Disabling node ping

To disable node ping, you simply set the node ping timeout value to zero:

```
bigpipe timeout_node 0
```

WARNING

Node ping is the only form of verification that the BIG/ip Controller uses to determine status on node addresses. Should you turn node ping off while one or more node addresses are currently ***down****, the node addresses remain marked* ***down*** *until you turn node ping back on and allow the BIG/ip Controller to verify the node addresses again.*

timeout_svc

Description

Sets the amount of time that a specific node has to respond to a service check issued by the BIG/ip Controller. Note that there are three types of service checks, each of which is affected by the this setting:

- Service check where the BIG/ip Controller attempts to establish a connection to the service hosted by the node
- Extended content verification where the BIG/ip Controller requests specific content from the node
- Extended application verification where the BIG/ip Controller executes an external service check program that verifies whether or not specific content is available on the node

If a node fails to respond to any type of service check within the specified time, the BIG/ip Controller assumes that the service is down and no longer sends requests to the node. If the node responds to the next service check, or to subsequent service checks, the BIG/ip Controller marks the node *up*, and resumes sending requests to the node.

WARNING

*If node ping is turned off (***bigd -n***) and values have not been set (value of 0) for* **timeout_svc** *and* **tping_svc** *for any services on a particular node, then BIGd does attempt detect the status of that node.*

The **timeout_svc** default is for each port is set to **0**, which disables service checks on the port.

Example

```
bigpipe timeout_svc 80 120
bigpipe timeout_svc 23 240
bigpipe timeout_svc
```

The following output is then displayed:

```
port 23 timeout after 240 seconds
```

```
port 80 timeout after 120 seconds
```

Note that the BIG/ip Controller monitors only those services which are specifically listed.

Syntax

```
bigpipe timeout_svc
bigpipe timeout_svc <port>
bigpipe timeout_svc <port> <seconds>
bigpipe timeout_svc <port> 0
```

Displaying service check timeouts

Use the following syntax to display the current service check timeout settings for all ports:

```
bigpipe timeout_svc
```

Use the following syntax to display the current service check timeout setting for a specific port:

```
bigpipe timeout_svc <port>
```

Setting the service check timeout

Use the following syntax to set the service check timeout for a specific node port. Note that this setting applies to all nodes that use the node port.

```
bigpipe timeout_svc <port> <seconds>
```

To disable service check on a specific port, use the above command, but set the <seconds> parameter to zero:

```
bigpipe timeout_svc <port> 0
```

tping_node

Description

Sets the interval (in seconds) at which a BIG/ip Controller issues a ping to each server managed by the BIG/ip Controller. If a specific server responds to the ping within a set time, the nodes hosted by that server are marked *up* and the BIG/ip Controller sends connections to the nodes hosted by that server. If a server fails to respond to a ping within the specified time, the BIG/ip Controller assumes that the nodes hosted by the server are no longer available, and it marks the nodes down.

Note that the **timeout_node** setting determines the number of seconds that a server has to respond to the ping issued by the BIG/ip Controller.

The default setting for **tping_node** is 5 seconds.

Example

The following command sets **tping_node** to be 10 seconds.

```
bigpipe tping_node 10
bigpipe tping_node
```

The following output is then displayed:

```
tping_node 10
```

Syntax

```
bigpipe tping_node
bigpipe tping_node <seconds>
```

Displaying the current node ping setting

Use the following syntax to display the current node ping setting:

```
bigpipe tping_node
```

Setting a node ping interval

Use the following syntax to set the number of seconds which a server has to respond to a ping issued by the BIG/ip Controller:

```
bigpipe tping_node <seconds>
```

Disabling node ping

To turn node ping off, simply set the interval to 0 seconds as shown below:

```
bigpipe tping_node 0
```

tping_svc

Description

Sets the interval (in seconds) at which BIG/ip Controller issues a service check to one or more specific nodes included in the configuration. There are three types of service check, each of which is affected by this setting:

- Service check where the BIG/ip Controller attempts to establish a connection to the service hosted by the node
- Extended content verification where the BIG/ip Controller requests specific content from the node
- Extended application verification where the BIG/ip Controller executes an external service check program that verifies whether or not specific content is available on the node

If a node fails to respond to a service check within the time specified by the **timeout_svc** setting, the BIG/ip Controller marks the node *down*, and no longer routes client requests to it.

WARNING

*The bigd daemon does not attempt to detect the status of a node if node ping is turned off (***bigpipe tping_node 0***), and the* **timeout_svc** *and* **tping_svc** *values are set to* **0** *for any services on the particular node.*

The **tping_svc** default is set to **0**, which disables service checks.

Example

```
bigpipe tping_svc 23 60
bigpipe tping_svc 80 15
bigpipe tping_svc
```

The following output is then displayed:

```
port 23 ping every 60 seconds
port 80 ping every 15 seconds
```

Note that the BIG/ip Controller monitors only those services which are specifically listed.

Syntax

```
bigpipe tping_svc
bigpipe tping_svc <port> <seconds>
bigpipe tping_svc <port> 0
```

Displaying the current service check interval

Use the following syntax to display the intervals at which the BIG/ip Controller issues service checks to all nodes configured for service check:

```
bigpipe tping_svc
```

Setting global service check intervals for a node port

Use the following syntax to set a service check interval for a specific node port.

```
bigpipe tping_svc <port> <seconds>
```

Use the following syntax to turn service check off for specific a node port.

```
bigpipe tping_svc <port> 0
```

treaper

Description

Sets the expiration time for idle connections on a specific virtual port. An idle connection is one in which no data has been received or sent for the number of seconds specified by the **treaper** command. The **treaper** default value is 0 seconds, meaning that no idle connections are terminated. For **treaper** to be effective, you should set its value to be greater than the configured timeout for the service daemons installed on your nodes.

Example

```
bigpipe treaper 23 600
bigpipe treaper 80 1200
bigpipe treaper
```

The following output is then displayed:

```
connections to port 23 reaped if idle for longer than 600 seconds
connections to port 80 reaped if idle for longer than 1200 seconds
```

The BIG/ip Controller terminates idle connections only for those ports that are specifically listed.

Syntax

```
bigpipe treaper
bigpipe treaper <port> <seconds>
bigpipe treaper <port> 0
```

Displaying the current inactive connection timeout

Use the following syntax to display the current number of seconds that connections are allowed to remain idle before being dropped:

```
bigpipe treaper
```

Setting the inactive connection timeout for a virtual port

Use the following syntax to set an inactive connection timeout for one or more virtual ports:

```
treaper <port> <seconds>
```

To turn inactive connection timeout off, use the above command, setting the number of seconds to zero:

`treaper <port> 0`

Note

Typical default settings include 120s for 25/SMTP, 120s for 80/www, 300-600 for 20/ftp-data and 21/ftp-data.

udp

Description

The **udp** command enables UDP traffic on virtual ports and sets UDP persistence for those ports. UDP traffic is enabled only when the persistence is set to a value greater than 0 (zero). Setting persistence to 0 disables UDP on that port.

By default, UDP is disabled on all ports.

Persistence forces UDP packets that have the same source address and virtual server as prior UDP packets to use the same node as used by the prior UDP packets for the specified period. Note that certain system control variables affect the behavior of the persistence timer, as well as the behavior of persistence itself. Refer to Appendix C for more information.

Example

```
bigpipe udp 53 300
bigpipe udp
```

The following output is then displayed:

```
port 53 idle udp connections expire after 300 seconds
```

The BIG/ip Controller allows persistence only for those services specifically listed.

Syntax

```
bigpipe udp
bigpipe udp <port>
bigpipe udp <port> <seconds>
bigpipe udp <port> 0
```

Displaying UDP settings

Use the following syntax to display the persistence settings for all ports that allow UDP:

```
bigpipe udp
```

Use the following syntax to display the persistence setting for a specific virtual port that allows UDP:

```
bigpipe udp <port>
```

Setting UDP persistence for a virtual port

Use the following syntax to set persistence on one or more virtual ports, where the **<seconds>** parameter is the number of seconds for which the BIG/ip Controller maintains persistence information for a particular session.

```
bigpipe udp <port> <seconds>
```

To turn UDP persistence off for a virtual port, use the above command, setting the **<seconds>** parameter to zero:

```
bigpipe udp <port> 0
```

-v

Description

Displays version number of the BIG/pipe command utility. For example:

```
bigpipe: version 1.8
```

Syntax

```
bigpipe -v
```

version

Description

Displays the version number of the BIG/ip Controller's operating system. For example:

```
BIG/ip: version 1.8
```

Syntax

bigpipe version

vip

Description

The **vip** command creates, deletes, and displays information about ***virtual servers***. This command also allows you to set properties on a virtual server. A virtual server defines the relationships between an externally visible IP address that clients use to connect to your site, and the internal IP addresses of individual nodes that actually provide services for your site. The **bigpipe vip** command supports a variety of parameters.

Syntax

```
bigpipe vip
bigpipe vip <virt addr>:<port>... <virt addr>:<port>
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port>
bigpipe vip <virt addr>:<port> enable
bigpipe vip <virt addr>:<port> disable
bigpipe vip <virt addr>:<port>... <virt addr>:<port> delete
bigpipe vip <virt addr>... <virt addr> limit <limit>
bigpipe vip <virt addr>:<port>... <virt addr>:<port> limit 0
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port> special ssl <persistence timeout> \
   <inactive connection timeout>
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port> special ssl 0 0
bigpipe vip <virt addr>:<port> <bitmask> define <node addr>:<port>\
   <node addr>:<port>
bigpipe vip <virt addr>:<port> netmask <netmask> define \
   <node addr>:<port>... <node addr>:<port>
bigpipe vip <virt addr>:<port> broadcast <broadcast> define \
   <node addr>:<port>... <node addr>:<port>
bigpipe vip <virt addr>:<port> netmask <netmask> broadcast
   <broadcast> define <node addr>:<port>... <node addr>:<port>
bigpipe vip <virt addr>... <virt addr>
bigpipe vip <virt addr> enable
bigpipe vip <virt addr> disable
```

Displaying information about virtual servers

Use the following syntax to display information about all virtual servers included in the configuration:

```
bigpipe vip
```

Use the following syntax to display information about one or more virtual servers included in the configuration:

```
bigpipe vip <virt addr>:<port>... <virt addr>:<port>
```

The command display information such as the nodes associated with each virtual server, the nodes' status, and the current, total, and maximum number of connections managed by the virtual server since the BIG/ip Controller was last rebooted, or since the BIG/ip Controller became the active unit (redundant configurations only).

Displaying information about virtual addresses

You can also display information about the virtual addresses that host individual virtual servers. Use the following syntax to display information about one or more virtual addresses included in the configuration:

```
bigpipe vip <virt addr>... <virt addr>
```

The command display information such as the virtual servers associated with each virtual address, the status, and the current, total, and maximum number of connections managed by the virtual address since the BIG/ip Controller was last rebooted, or since the BIG/ip Controller became the active unit (redundant configurations only).

Defining a virtual server

Virtual servers are port-specific, and if you are configuring a site that supports more than one service, you need to configure one virtual server for each service offered by the site. Use the following syntax to define an individual virtual server and the node or nodes to which the virtual server maps:

```
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port>
```

For example, the following command configures a virtual server that uses three nodes. In the example, two of the nodes do not use port 80, the standard HTTP port. Node port numbers do not necessarily have to match the virtual server's port number.

```
bigpipe vip 210.12.140.100:80 define 192.168.11.22:80 \
   192.158.11.23:8080 192.168.11.23:8050
```

Note that if you want to add or remove a node from a virtual server, you must redefine the virtual server. You cannot add or remove individual nodes from a virtual server mapping without redefining the virtual server itself.

The following example shows a similar definition where host names are used in place of IP addresses, and service names are used in place of port numbers. Note that if you use service names, the default port number associated with that service is used.

```
bigpipe vip www.SiteOne.com:http define NodeOne:http NodeTwo:http
   NodeThree:http
```

If you are using non-default ports to host a specific service, you should use the port number in the definition rather than the service name.

Setting a user-defined netmask and broadcast

The default netmask for a virtual address, and for each virtual server hosted by that virtual address, is 255.255.255.0. The default broadcast is automatically determined by the BIG/ip Controller, and it is based on the virtual address and the current netmask. You can override the default netmask and broadcast for any virtual address. All virtual servers hosted by the virtual address inherently use the netmask and broadcast of the virtual address, whether they are the defaults or they are user-defined.

Use the following syntax to set a user-defined netmask when you define the virtual server:

```
bigpipe vip <virt addr>:<port> netmask <netmask> define \
   <node addr>:<port>... <node addr>:<port>
```

Use the following syntax to set a user-defined broadcast address when you define the virtual server:

```
bigpipe vip <virt addr>:<port> broadcast <broadcast> define \
   <node addr>:<port>... <node addr>:<port>
```

Note that if you want to use a custom netmask and broadcast, you define both when defining the virtual server:

```
bigpipe vip <virt addr>:<port> netmask <netmask> \
   broadcast <broadcast> define <node addr>:<port>... \
   <node addr>:<port>
```

Note

For most configurations, the BIG/ip Controller correctly calculates the broadcast based on the IP address and the netmask, and a user-defined broadcast address is not necessary.

Again, even when you define a custom netmask and broadcast in a specific virtual server definition, the settings apply to all virtual servers that use the same virtual address. The following sample command shows a user-defined netmask and broadcast:

```
bigpipe vip www.SiteOne.com:http netmask 255.255.0.0 \
   broadcast 10.0.140.255 define NodeOne:http NodeTwo:http
```

Setting properties on a virtual server

You can set the following properties on a virtual server:

- A connection limit
- An SSL persistence timeout and an SSL inactive connection timeout

Setting a connection limit

The default setting is to have no limit to the number of concurrent connections allowed on a virtual server. You can set a concurrent connection limit on one or more virtual servers using the following syntax:

```
bigpipe vip <virt addr>:<port>... <virt addr>:<port> limit <limit>
```

The following example shows two virtual servers set to have a connection limit of 5000 each:

```
bigpipe vip www.SiteOne.com:http www.SiteTwo.com:ssl limit 5000
```

To turn the limit off, set the **<limit>** variable to zero:

```
bigpipe vip <virt addr>:<port>... <virt addr>:<port> limit 0
```

Defining SSL persistence settings

You can turn on SSL persistence for a virtual server when you define the virtual server. The command includes parameters for setting the timeout for an SSL session ID, as well as an inactive connection timeout for SSL connections:

```
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port> special ssl <persistence timeout> \
   <inactive connection timeout>
```

Note that if you want to change SSL settings on an existing virtual server, you must redefine the virtual server, including the nodes to which the virtual server maps and the SSL persistence settings. To turn SSL persistence off, use the above command, setting both the **<persistence timeout>** and **<inactive connection timeout>** parameters to **0**:

```
bigpipe vip <virt addr>:<port> define <node addr>:<port>... \
   <node addr>:<port> special ssl 0 0
```

The following example shows a virtual server set to use SSL persistence where the SSL session ID is maintained by the BIG/ip Controller for 36000 seconds, and inactive SSL connections are dropped after 60000 seconds:

```
bigpipe vip 210.12.140.11:443 define NodeOne:ssl NodeTwo:ssl \
   special ssl 36000 60000
```

Removing and returning a virtual server to service

You can remove an existing virtual server from network service, or return the virtual server to service, using the **disable** and **enable** keywords. When you disable a virtual server, the virtual server no longer accepts new connection requests, but it allows current connections to finish processing before the virtual server goes down. Use the following syntax to remove a virtual server from network service:

```
bigpipe vip <virt addr>:<port> disable
```

Use the following syntax to return a virtual server to network service:

```
bigpipe vip <virt addr>:<port> enable
```

Removing and returning a virtual address to service

You can remove an existing virtual address from network service, or return the virtual address to service, using the **disable** and **enable** keywords. Note that when you enable or disable a virtual address, you inherently enable or disable all of the virtual servers that use the virtual address.

```
bigpipe vip <virt addr> disable
```

Use the following syntax to return a virtual address to network service:

```
bigpipe vip <virt addr> enable
```

Deleting a virtual server

Use the following syntax to permanently delete one or more virtual servers from the BIG/ip Controller configuration:

```
bigpipe vip <virt addr>:<port>... <virt addr>:<port> delete
```

C

BIG/ip System Control Variables

Setting BIG/ip system control variables

There are many system control variables that provide information about the BSDI system, or that control certain functionality which is not developed by F5 Labs. Table 3.1 outlines the system control variables which you can change to affect certain BIG/ip Controller features. Note that these variables use the standard toggle on and off setting, where you set the variable value to 1 (one) for on, and set the variable value to 0 (zero) for off. System control variable settings are stored in the */etc/rc.sysctl* file. You can add a system control variable line to the file, or you can execute a **`sysctl`** command directly on the command line.

Displaying current system control variable settings

To display the settings of all system control variables, use the following syntax:

```
sysctl -a
```

To display the current setting for an individual variable, use the following command syntax:

```
sysctl <variable name>
```

Setting a system control variable

Use the following syntax to set a system control variable, where **`<setting>`** accepts only two values: a 0 to disable the variable, and a 1 to enable the variable:

```
sysctl -w <variable name>=<setting>
```

For example, the following command turns Transparent Node Mode on.

```
sysctl -w bigip.bonfire_mode=1
```

To turn Transparent Node Mode off, you would issue the following command:

```
sysctl -w bigip.bonfire_mode=0
```

Variable	Description	Default
bigip.vipnoarp	Prevents the BIG/ip Controller from issuing ARP requests when the unit is rebooted. This is useful for configurations that contain 1,000 or more virtual servers.	**0**
bigip.bonfire_mode	When this variable is on, the BIG/ip Controller operates in Transparent Node Mode, where it can perform load balancing on routers and router-like devices, such as transparent firewalls.	**0**
net.inet.ip.forwarding	Turns on IP forwarding for traffic not destined to a virtual server or NAT address.	**0**
bigip.webadmin_port	When running the administrative web server on a non-default port, turn this variable on to allow traffic on that port to be processed by the BIG/ip Controller.	**0**

Variable	Description	Default
`bigip.persist_time_used_as_limit`	When set to 0, this variable forces the persistence timer to be reset on each packet for persistent sessions. Normally the timer starts when a connection is first made, and subsequent connections go to the same node until the timeout expires. In this mode, the timer does not expire as long as there is traffic. Note that the variable does not affect SSL session ID persistence. For SSL persistence, the timer is always reset on each packet.	`1`
`bigip.persist_on_any_vip`	When active, the BIG/ip Controller sends all persistent connections from the same client to the same node address, regardless of which virtual server hosts the persistent connections. This mode is not applicable to SSL session ID persistence.	`0`
`bigip.persist_on_any_port_same_vip`	When active, this mode requires that the BIG/ip Controller send all persistent connections going to a particular virtual address from the same client to the same node address, regardless of which virtual server associated with the virtual address hosts the persistent connections. This mode is not applicable to SSL session ID persistence.	`0`

Table 3.1 *System control variables for the BIG/ip Controller*

D

Services and Port Index

.

Service	Port	Description
tcpmux	1	# TCP port multiplexer (RFC1078)
echo	7	
discard	9	
systat	11	# Active Users
daytime	13	
chargen	19	
ftp-data	20	
ftp	21	
ssh	22	# Secure shell
telnet	23	
smtp	25	# sendmail
time	37	# timserver
nameserer	42	# name, IEN 116
ni-ftp	42	# NI FTP
whois	43	# nicname
xns-time	52	# XNS Time Protocol
domain	53	# name-domain server
xns-ch	54	# XNS Clearinghouse
xns-auth	56	# XNS Authentication
xns-mail	58	# XNS Mail
tacacs-ds	65	# TACACS-Database Service
sql*net	66	# Oracle SQL*NET
bootps	67	# bootp/dhcp server
bootpc	68	# bootp/dhcp client
tftp	69	
gopher	70	
finger	79	
http	80	# www
npp	92	# Network Printing Protocol
objcall	94	# Tivoli Object Dispatcher
hostnames	101	# usually from sri-nic
tsap	102	# part of ISODE.
csnet-ns	105	# Mailbox Name Nameserver
rtelnet	107	# Remote Telnet Service
snagas	108	# SNA Gateway Access Server

Service	Port	Description
pop2	109	# old pop port
pop	110	# pop3 postoffice
ident	113	# auth tap authentication
sftp	115	
sqlserv	118	# SQL Services
nntp	119	# USENET News Transfer Protocol
ntp	123	# network time protocol
ingres-net	134	# INGRES-NET Service
netbios-ns	137	# SMB Name Service (SAMBA)
netbios-ssn	139	# SMB Session Service (SAMBA)
imap2	143	# Interactive Mail Access Protocol v2
iso-tp0	146	# ISO-IP0
iso-ip	147	# ISO-IP
sql-net	150	# SQL-NET
bftp	152	# Background File Transfer
sgmp	153	
sqlsrv	156	# SQL Service
sgmp-traps	160	
snmp	161	
snmp-trap	162	
print-srv	170	# Network PostScript
bgp	179	# Border Gateway Protocol
gacp	190	# Gateway Access Control Proto
prospero	191	# Prospero Directory Service
irc	194	# Internet Relay Chat Protocol
smux	199	
ipx	213	
dbase	217	# dBASE Unix
imap3	220	# Interactive Mail Access Protocol v3
pdap	344	# Prospero Data Access Protocol
ulistserv	372	# Unix Listserv
hp-collector	381	# hp perf data collector
hp-managed-node	382	# hp perf data managed node
hp-alarm-mgr	383	# hp perf data alarm manager
unidata-ldm	388	# Unidata LDM Version 4
ldap	389	# Lightweight Directory Access

Service	Port	Description
synotics-relay	391	# SynOptics SNMP Relay Port
synotics-broker	392	# SynOptics Port Broker Port
netware-ip	396	# Novell Netware over IP
prm-sm	408	# Prospero Resource Manager
prm-nm	409	# Prospero Resource Manager
rmt	411	# Remote MT Protocol
infoseek	414	
https	443	# SSL-based http
snpp	444	# Simple Network Pager Protocol
biff	512	# comsat
login	513	
shell	514	# no passwords used
printer	515	# line printer spooler
talk	517	
ntalk	518	
route	520	# router routed
timed	525	# timeserver
conference	531	# chat
netnews	532	# readnews
klogin	543	# Kerberos rlogin
kshell	544	# Kerberos remote shell
gii	611	# Gated Interactive Interface
doom	666	# doom Id Software
flexlm	747	# Flexible License Manager
kerberos-adm	749	# kerberos administration
kerberos	750	# Kerberos (server) tcp
kpasswd	751	# Kerberos "passwd"
krbupdate	760	# Kerberos registration
webster	765	
webster	765	
phonebook	767	# phone
rpasswd	774	
socks	1080	# SOCKS
kpop	1109	# Kerberos pop
prm-sm-np	1402	# Prospero Resource Manager
prm-nm-np	1403	# Prospero Resource Manager

Service	Port	Description
ms-sql-s	1433	# Microsoft-SQL-Server
ms-sql-m	1434	# Microsoft-SQL-Monitor
watcom-sql	1498	# Watcom-SQL
ingreslock	1524	
dirsrv	1525	# Archie directory service
prospero-np	1525	# Prospero Dir Service Non-priv
pdap-np	1526	# Prospero Data Access Proto
tlisrv	1527	# oracle
coauthor	1529	# oracle
radius	1645	
snmp-tcp-port	1993	# cisco SNMP TCP port
gdp-port	1997	# cisco Gateway Discovery Proto
eklogin	2105	# Kerberos encrypted rlogin
ccmail	3264	# cc:mail/lotus
aol	5190	# America-Online
amanda	10080	# regular BSD auth amanda
kamanda	10081	# Kerberos auth amanda
isode-dua	17007	

Index

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

V

W